University of Calabria
Department of Ingegneria Informatica, Modellistica, Elettronica e Sistemistica - DIMES

Sensor Selection and Reconciliation Methodologies for Fault-Tolerant Estimation

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

Doctoral Programme in
Information and Communication Technologies

XXXVII Cycle

Doctoral Dissertation of:
Franco Angelo Torchiaro

Supervisor:
Prof. Alessandro Casavola

Co-Supervisor:
Prof. Francesco Tedesco

Supervisor of the Doctoral Programme:
Prof. Giancarlo Fortino

January, 2025

Preface

This thesis embraces mostly of the efforts that I have invested during the last three years as a Ph.D. student at Università della Calabria. I have been working under the expert supervision of Prof. Alessandro Casavola, who is also the leader of the research group I am part of. A facilitator of opportunities, he has taught me how helpful can be to look at complicated problems with passionate eyes. As a complementary actor, my other advisor, Prof. Francesco Tedesco, tough me how to be precise and meaningful in my math, and has, more than once, been a savior when the details weren't working out. I appreciate both my advisors' patience in my learning process.

My research focuses on modeling, estimation and control tools, a field entirely unfamiliar until my second year of academic study. At that time, the passionate teaching of Professor Domenico Famularo in the course "Basics of Automation" ignited a spark. I recall saying, "This topic seems very interesting!". Those famous words led me to pursue a master's degree and, ultimately, a Ph.D. in this field.

During the three years of research, I had the wonderful opportunity of being "initiated" into the world of research. This, sometime, involved juggling multiple projects simultaneously, a challenging but rewarding experience. Each project exposed me to diverse people and places that, in a way or another, changed and improved the way I look at things. Among these experiences, my six-month stay at the Department of Electrical and Systems Engineering, Washington University in St. Louis, Missouri, stands out. I had the opportunity to collaborate with extraordinary individuals, particularly my advisor, Prof. Bruno Sinopoli. He was an impressive guide and source of inspiration.

Countless people have directly and indirectly contributed to my research, and this work is no exception. A major thanks goes to both Prof. Yilin Mo and Prof. Mehdi Forouzanfar for the time they have dedicated in reviewing this work and for their constructive comments. I will not, ever, be thankful enough to Ayman, an invaluable colleague and friend. Despite my innate ability to complain, he has always been a source of unwavering support. I am thankful to Cristian, Giuseppe L., and Stefano as well. They have always been priceless life companions.

On the colleagues-side of this acknowledgments I put all the fellows of Applicon: Francesco, Gianfranco, Gianni and Marco. I shared with them all the pain of working with prototypes.

On the friends-side of this list, Alex, Giuseppe N., Pasquale, and Rosario deserve a special mention as well. They have always been a breath of fresh air.

I have tried to translate in simple words the infinite gratitude I have and will always have to my sisters Antonella and Giovanna and my parents, Franco and Giuseppina, for being my fixed point in my life. Obviously, I failed.

Franco Angelo Torchiaro
Cosenza
January, 2025

Abstract

*T*his dissertation delves into the development of novel state estimation architectures and sensor selection methodologies to enhance fault tolerance in state estimation schemes. The research addresses a twofold objective: designing fault-hiding architectures, and determining the optimal number/placement of sensors to efficiently support these architectures.

The first objective advances the state-of-the-art in fault-tolerant estimation schemes by introducing novel architectures based on the Sensor Reconciliation methodology. Both centralized and distributed approaches are explored. A centralized method is proposed that combines a Luenberger observer and a reconciliator unit into a merged component that is used to identify and mitigate faults. Recognizing the limitations of centralized solutions for large-scale systems, a distributed sensor reconciliation architecture is also proposed, based on the decomposition of a centralized steady-state Kalman filter. This latter approach allows for distributed state estimation by fusing local measurements and estimates from neighboring agents. Both methods leverage sensor redundancy to enhance robustness against sensor faults.

Secondly, the dissertation explores the optimization of sensor selection. A Mixed-Integer Semidefinite Programming (MISDP) framework is defined and presented to systematically optimize sensor selection and design the corresponding L_1 optimal observer. This approach minimizes the number of sensors required for full state reconstruction while optimizing specific state reconstruction metrics to enhance efficiency, reliability, and robustness. Theoretical validation of the sensor placement's optimality is also provided.

The effectiveness of all proposed methodologies is rigorously assessed through extensive simulations. In particular, a generic road network traffic flow monitoring scenario serves as a practical testbed to demonstrate the potential applications of this research.

Abstract (Italian translation)

*I*l lavoro di tesi si focalizza sullo sviluppo di nuove architetture per la stima resiliente e la selezione di sensori, con lo scopo di individuare schemi di ricostruzione dello stato che garantiscano una certa tolleranza a guasti improvvisi. Più in dettaglio, la ricerca si pone un duplice obiettivo: progettare architetture in grado di mascherare guasti di sensori e, allo stesso tempo, determinare il numero/posizionamento ottimale di tali sensori così da supportare efficacemente il processo di stima.

Al fine di raggiungere il primo obiettivo, vengono proposte due nuove architetture, entrambe basate su una tecnica resiliente ai guasti nota in letteratura come "sensor reconciliation". A tal proposito, ad essere esplorati sono sia approcci centralizzati che distribuiti. Il primo metodo ad essere presentato è caratterizzato da un approccio puramente centralizzato che combina un osservatore di Luenberger e un'unità di riconciliazione in un unico componente in grado di identificare e mitigare i guasti. Riconoscendo i limiti delle soluzioni centralizzate quando applicate a sistemi complessi o su larga scala, viene anche proposta un'architettura di riconciliazione dei sensori distribuita, basata sulla decomposizione di un filtro di Kalman "steady-state" centralizzato. Quest'ultimo consente di stimare lo stato di un sistema in modo puramente distribuito fondendo misurazioni locali con stime condivise da agenti/sensori vicini. Entrambi i metodi sfruttano sensori ridondanti per migliorare la robustezza della struttura di stima a fronte di guasti imprevisti.

In secondo luogo, il lavoro di tesi esplora l'ottimizzazione della selezione dei sensori. A tal proposito, viene definito e presentato un framework che sfrutta la programmazione semidefinita a variabili mixed-integer (MISDP) per ottimizzare sistematicamente la selezione dei sensori e progettare il corrispondente osservatore L_1 ottimale. Questo approccio consente di minimizzare il numero di sensori necessari per la ricostruzione completa dello stato ottimizzando, allo stesso tempo, metriche specifiche in modo tale da migliorare efficienza, affidabilità e robustezza del processo di stima. Vengono inoltre forniti dettagli sulla dimostrazione teorica dell'ottimalità del posizionamento dei sensori.

L'efficacia di tutte le metodologie proposte viene rigorosamente valutata attraverso estese simulazioni. In particolare, uno scenario generico di monitoraggio del flusso di traffico di una rete stradale funge da banco di prova comune per dimostrare le potenzialità applicative delle soluzioni proposte.

Contents

List of Figures

List of Tables

Acronyms

AFTC Active Fault Tolerant Control. 4, 5

DSR Distributed Sensor Reconciliation. vi, vii, 7, 45, 46, 52, 53, 57–59, 86

FDI Fault Diagnosis and Isolation. 4, 5

FTC Fault Tolerant Control. 2, 4, 5

GA Genetic Algorithm. viii, 27, 29, 34, 35, 81, 82

KF Kalman Filter. 45, 86

LMI Linear Matrix Inequality. 7, 19, 23, 24

LPV Linear Parameter-Varying. 2, 5, 15–17, 21–23, 44

LSS Large-Scale System. 4, 7

LTI Linear Time Invariant. 5, 7–9, 13, 26, 31, 36, 45, 46, 55, 68, 75, 86

MAE Mean Absolute Error. 38

MAS Multi-Agent System. 4, 6, 45

MISDP Mixed-Integer semidefinite Programming. 8

MSIF Multisensor Integration and Fusion. 6

PEM Prediction Error Minimization. 40

RB Reconfiguration Block. 1, 4, 5

SA Simulated Annealing. viii, 27–29, 34, 35, 40, 81, 82

SR Sensor Reconciliation. 1, 2, 4

VS Virtual Sensor. 5

WSN Wireless Sensor Network. 7, 30, 44

Colophon

This document was created using the XeTeX typesetting engine derived from Donald Knuth's original TEXsoftware and the memoir class created by Peter Wilson. The document was compiled using the corresponding LaTeX compiler XelateX. The body text is set 11pt with standard Latin Modern font.

Introduction 1

1.1 The Sensor Reconciliation and Selection Problems

Sensor Selection and Sensor Reconciliation problems, while distinct, are intrinsically linked. Within the sensor reconciliation paradigm, observer design and state estimation theory serve as powerful tools for fault identification and accommodation. As a consequence, a wise sensor selection can significantly enhance estimation performance and, consequently, fault detection capability. This section examines the two problems through a common lens, identifying their points of contact and exploring how addressing one problem can be beneficial for the other.

1.1.1 The Sensor Reconciliation Problem

When sensors fail, the lack of reliable data or its complete loss can lead to serious malfunctions, such as loss of stability or tracking performance in control loops [1], [2], or make the entire architecture unreliable for monitoring applications. To prevent such scenarios, a possible approach involve the introduction of Sensor Reconciliation (SR) units. These units are usually inserted between the sensors and the downstream modules that are in charge to use sensor data. From a general point of view SR units fall into the broader category of Reconfiguration Block (RB) [3]. In fact, when faults are detected, SR units, provided certain assumptions, undergo a self reconfiguration in such a way to accommodate the detected faults, thus providing reliable sensor data. In particular, RBs whose objective is to recover from sensor faults are generally referenced to as *Virtual Sensors*. *Virtual Sensors* are inspired by observer design methods and aim at estimating fault system states and outputs despite the sensor faults. The idea of using the estimators for fault identification and accommodation has already inspired several fault-tolerant sensor control approaches prior to the emergence of fault hiding approaches. Indeed, it is well-known that the recovery of systems with sensor faults can be performed by observers or filters which use the analytical or physical redundancies [4].

The SR units use this concept as well in combination with a *re-conciliator Unit*. The key idea is to estimate the state of the faulty system via a reconfigurable adaptive observer. The faulty output and the estimated output is fed to

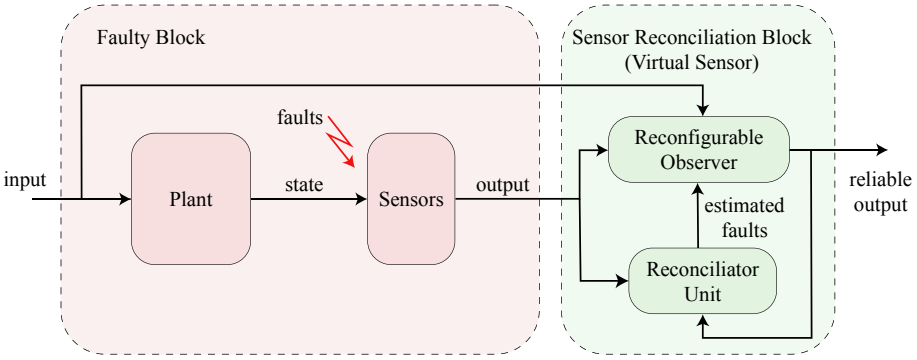

FIGURE 1.1: Sensor Reconciliation Basic scheme.

a reconciliator unit for fault detection and isolation. The estimated faults are then used to reconfigure the observer and accommodate the fault. A graphical representation that explains the SR philosophy is given in Figure 1.1.

Many SR approaches based on Figure 1 are often proposed in the literature as part of a Fault Tolerant Control (FTC) scheme and coupled with traditional controllers. Such a choice avoids the usage of complex control reconfiguration strategies to accommodate sensor faults. In this respect, relevant contributions include [5], [6] where the sensor information are fused in a decentralized way by exploiting local estimators. Another class of SR FTC based strategies is considered in [7], [8], where a switching mechanism involving sensors and related observers is exploited to implicitly detect the healthy components of the system. The estimates provided by the observers are compared at each sampling time by a switching logic that allows one to select the sensors-observer pair with the smallest estimation error. More recently, SR schemes have been proposed that involve Linear Parameter-Varying (LPV) [9] and Linear Fractional Transformation [10] Unknown Input Observers where the estimation of some fault parameters is carried out via parity space analysis [11]–[13]. In particular, an output predictor is build based on state-space matrices and the difference between the expected outputs and the measured ones along a moving past window is used to detect faults and apply countermeasures.

1.1.2 The Sensor Selection Problem

In a general sensor reconciliation architecture a given set of sensors provides output data used by a virtual sensor to estimate the internal state of the monitored system with the aim of detecting anomalies and faults. When multiple sensors are available for estimation, an intelligent sensor selection significantly enhances estimation performance and, consequently, fault detection capability. Determining optimal sensor selections for a specific objective is an NP-hard combinatorial optimization problem, typically requiring exhaustive enumeration of all possible configurations [14].

Several more practical solutions have been proposed by considering the problem from a system observability perspective. Observability is closely tied to state estimation performance through metrics derived from the observability Gramian.

The interpretation of the observability Gramian centers on the quadratic form associated with the energy of the free response induced by the initial state: a larger value indicates a more observable initial state. Practically, this implies that cheaper sensing is feasible when an observability-Gramian-based metric suggests an energy increase independent of the initial state [15]. Several examples of such metrics are drawn from [14], [16]–[18] and listed below:

- *rank*: the rank of Gramian matrix determines the dimension of the observable subspace;
- *trace*: the trace of Gramian matrix is directly related to the average energy and can be interpreted as the average observability in all directions of the state;
- *condition number*: the condition number of the Gramian measures how balanced the observability is among all modes. It grows unbounded if the system is unobservable;
- *eigenvalues*: the minimum (maximum) eigenvalue of the Gramian represents the smallest (largest) energy associated to the free response required to estimate a unit-norm initial state; the minimum eigenvalue can be zero for unobservable systems.
- *determinant*: whenever the Gramian is invertible, the determinant of the observability Gramian is proportional to the volume of the ellipsoid containing the initial states that can be observed with unit or less estimation energy;
- $H2$-*norm*: The intuition behind this metric is that more potent sensors usually output stronger signals, and this potency is captured by the $H2$-norm.
- *logdet*: this metric can be intended as a volumetric variant of the $H2$-norm.

In light of this premise, optimal sensor selection involves identifying a subset of potentially redundant sensors from a larger set of candidate sensors to preserve the structural observability of the monitored system. Authors of [19] address this task by proposing various Observability Gramian-based metrics that can be maximized through a greedy approach exploiting sub-modularity. This approach guarantees solutions with an upper-bounded optimality gap [20]. Other heuristics, such as simulated annealing [21], are employed to solve the problem under a predefined budget constraint on the maximum number of sensors. The results presented in [17] and [22] are also considered. The former proposes an end-to-end solution to the street sensor selection problem through road network modeling and observability measures. The latter addresses fault-tolerant estimation and the design of fault-tolerant sensor networks, introducing the concepts of redundant and minimal sensor sets organized into an automaton.

Extensively explored in the literature, various algorithms have been proposed to maximize $H2$ norm-related metrics, capitalizing on the assumption that more

powerful sensors yield stronger signals [14], [23], [24], [18]. Additionally, sensor selection has been approached from diverse angles, including optimization of Gramian-related metrics [25]–[28], linear-quadratic regulation [29]–[31], and security concerns [32], [33].

1.2 Motivation

This research delves into three primary areas: the design of a Sensor Reconciliation unit, its adaptation to distributed contexts, and the proper selection of sensors. Each area brings unique challenges and opportunities that have acted as inspiration for this study:

- **Reconciliation Problem**: One of the most common Fault Tolerant Control (FTC) approach is the Active Fault Tolerant Control (AFTC). This technique specifically aims to alleviate the effects of faults by modifying the baseline control strategy. Control reconfiguration is one such approach, which involves redesigning the controller for a reconfigured loop that utilizes all available (physical and analytical) redundancy. However, instead of using the same actuators and sensors, it adjusts the control loop to incorporate healthy components and designs a new controller for the modified loop. While effective, control reconfiguration is an invasive approach that requires frequent controller and loop modifications based on a Fault Diagnosis and Isolation (FDI) system indications. In the alternative approach explored in this dissertation the core idea involves inserting a reconfigurable block between the nominal controller (which remains unchanged) and the faulty plant. This RB, activated post-fault detection, ensures the recovery of system properties like stability and tracking. In numerous engineering applications this approach offers several advantages. In fact, it enables to achieve specific fault resilient objectives, while preserving the baseline controller. Furthermore, the inserted elements can operate on a minimum-change principle, becoming active only when necessary, and can simplify the certification process in highly regulated industries like energy, transportation, and aeronautics.

- **Distributed Problem**: In recent years, distributed state estimation for Multi-Agent System (MAS) or Large-Scale System (LSS) has gained significant attention due to its relevance in various applications, including cooperative control of vehicular platoons, synchronization of coupled oscillators, formation control of autonomous vehicles, and rendezvous of space shuttles [34]–[36]. For such systems a centralized architecture can be practically disadvantages or practically untractable due to the systems dimension and/or complexity. Moreover, in a distributed estimation context, one of the advantages of the fault-hiding FTC approach is the plug-and-play nature of RBs, making them particularly suitable for LSS.

- **Selection Problem**: Sensor placement optimization is crucial for effective state estimation, especially when estimating internal or hidden physical

quantities. Of particular interest for this research is the sensor selection problem applied to positive dynamical systems. Those systems provide a solid foundation for scenarios where observed variables are inherently non-negative. These systems model a wide range of real-world processes in fields such as biology, ecology [37], epidemiology [38], and networking [39]. Numerous studies have addressed observer design for linear positive (compartmental) systems, as evidenced by [40], [41], and [42]. A critical challenge in positive observation is constructing observers that guarantee non-negative estimates for intrinsically non-negative states. This is particularly important in real-world applications where negative estimates, such as negative volumes or concentrations, are meaningless. Unfortunately, as highlighted in [43], classical Luenberger observers can produce negative estimates, rendering them unsuitable for estimating inherently non-negative quantities.

Moved by the above considerations this research addresses a twofold question. Firstly, how can a fault-hiding architecture be designed and implemented in both a centralized and a distributed context? Secondly, what is the optimal number/locations of sensors to efficiently execute the first task? Prior to delving into the main thesis contents and results, a concise overview of the broader domains of *fault hiding* techniques and *distributed estimation* methodologies is presented.

1.3 Fault Hiding: a Brief Overview

What makes the Reconfiguration Blocks appealing in many applications is the possibility of achieving control objectives without modifying a controller which is already inserted in the loop. The term "*reconfiguration block*" was proposed by [44] for Fault Tolerant Control to designate an alternative to the most common Active Fault Tolerant Control approaches – namely fault accommodation, control reconfiguration – where the controller or the control loop used to be redesigned when a fault was detected and diagnosed by means of an FDI system. The fault tolerant control based on the insertion of RBs for recovering system properties after the fault detection and estimation is called *fault hiding* or *masking*. In particular, the canonical RB used for sensor fault hiding is the *Virtual Sensor*. Due to its efficacy and flexibility, the fault hiding approaches were extended to several classes of systems in addition to the Linear Time Invariant (LTI) ones, such as LPV [45], fuzzy [46], Lure [47], switched [48], Hammerstein–Wiener [49], and input-affine nonlinear [50] systems. Moreover, fault hiding based on virtual sensors and actuators has been successfully applied to different physical systems, e.g., hydraulic, aerospace, and power systems. However, conceiving the design of additional blocks in the control loops to achieve control goals that were not initially considered when the baseline controller was designed is prior to the concept of RB and fault hiding. Indeed, the applications of such an idea can be found within or without the fault-tolerant control body of knowledge.

For example, Multisensor Integration and Fusion (MSIF) [51]–[54] are techniques proposed to improve the accuracy of observations by combining the measurements from different sensors. The problem of integrating the information of fusion of individual sensor technologies to obtain a more accurate observation was first proposed in the seminal paper [53] concerning tactical warfare situations. However, solutions for this problem were only presented some years later by several researchers [54]–[57] using mainly stochastic and fuzzy techniques. Currently, most of the guidance and navigation control systems employ some kind of MSIF technique. In this context, the term "*virtual sensor*" was first used in one of the earliest papers [56] on MSIF.

1.4 Distributed Estimation and its Fault Tolerant Applications: : a Brief Overview

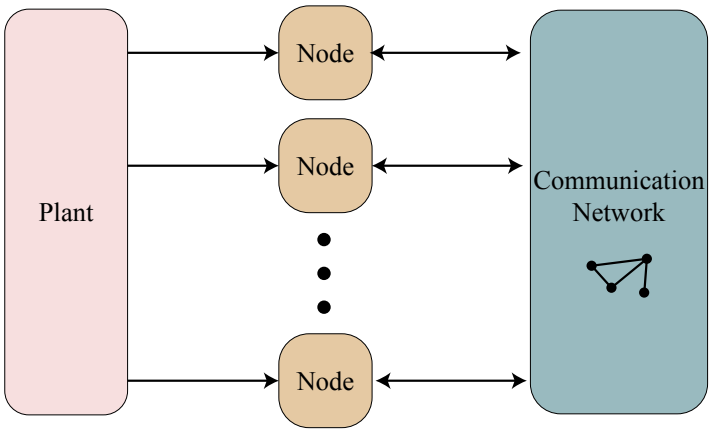

FIGURE 1.2: Distributed estimation scheme.

In the context of distributed state estimation of large-scale complex systems or Multi-Agent System (MAS), the objective is to design a distributed group of observers assigned to logical nodes such that each observer provides an estimate of the whole state vector of a dynamical system. Each observer has only access to local measurements and information received from a specific subset of other nodes. A communication network with a given specific topology enables the information flow among the nodes (this is a generic distributed estimation problem, as formulated in [58]). A graphical representation of such a scenario is given in Figure 1.2. For instance, authors in [59], [60] proposed distributed observer designs via the extension of the classical Kalman filter to a distributed observer network. In [61] and [62], the classical Luenberger observer is extended to a distributed estimation scheme. Moreover, in [63] a lossless decomposition of a steady state Kalman filter is derived for discrete autonomous systems. Consensus based algorithms have proved to be successful solutions in this field. In [64], a distributed estimation algorithm based on a moving horizon paradigm is introduced and in [65], necessary and sufficient conditions for distributed estimation in linear systems are derived by introducing augmented states satisfying

scalability conditions. Several fault hiding results for distributed systems do exists in literature as well. With particular focus on sensor faults, in [66], a two-layer distributed fault-hiding approach for linear LSSs subject to both sensor and actuator faults is designed. In [67] a state estimation methodology that uses a network of distributed observers, designed by decomposing the system according to an observability decomposition, for a class of Lipschitz nonlinear systems is proposed. In [68] a distributed virtual actuator and sensor fault tolerant consensus for Homogeneous Linear Multi-Agent Systems is proposed where the design is carried out exploiting Linear Matrix Inequality (LMI) tools. At the best of the author knowledge, currently an explicit distributed sensor reconciliation architecture is not available in literature.

1.5 Thesis Overview

In this section a wide analysis of the thesis structure is provided with a discussion on the contributions of the Ph.D. research achieved.

1.5.1 Thesis Outline

The main research findings discussed in this thesis are organized into three main chapters. The results are presented according to an incremental scheme as follows:

- **Chapter 2** introduces the fundamental mathematical concepts and definitions used through this work. Specifically, the state estimation problem is reviewed from both a centralized and distributed perspective. The basic concepts behind the Luenberger observer and Kalman filter are discussed and techniques for fault modeling are given. In conclusion, the Observability Gramian concept is defined as valuable tool for the definition of sensor selection metrics;

- **Chapter 3** presents the centralized design problem of determining a suitable fault-tolerant full state observer for large-scale Wireless Sensor Network (WSN) along with the concomitant sensor set selection for the effective state reconstruction in the presence of sensor faults/failures. Full details of the novel fault-tolerant sensor reconciliation design procedure are given demonstrating how the architecture is able to hide measurements that may be corrupted by sensor errors and provide reliable state estimates;

- **Chapter 4** presents a Distributed Sensor Reconciliation (DSR) methodology based on the decomposition of a centralized steady-state Kalman Filter, which is used to distributively estimate the state of a LTI plant in the presence of unpredictable sensor faults. In the distributed context it is supposed that individual local measurements may not ensure system observability, but their collective combination does. Theoretical guarantees and performance analysis through extensive simulations are provided;

- **Chapter 5** provides details about the concomitant sensor placement and observer design strategy designed for positive systems. The methodology proposed leverages Mixed-Integer semidefinite Programming (MISDP) techniques, providing a rigorous and systematic framework for optimizing the selection of sensors and the design of the corresponding optimal observer, according to appropriate metrics, in a unified manner;
- **Chapter 6** summarizes the results of this thesis and gives directions for future research.

1.5.2 Contribution

The main contributions of the research described in this Ph.D. thesis with respect to the Sensor Reconciliation and Selection problems are the following:

- A novel centralized Sensor Reconciliation Scheme applied to a compartmental traffic model for intelligent traffic flow estimation submitted and accepted for publication in future issues of the IEEE Transactions on Automation Science and Engineering [69];
- A novel Joint Sensor Selection and Observer design methodology for positive systems submitted and accepted for publication in IEEE Control System Letters [70];
- A novel Joint Sensor Selection and Observer design methodology for positive systems accepted for presentation at the IEEE Conference on Decision and Control (CDC) 2024 [71];
- A novel Distributed Sensor Reconciliation Scheme for LTI systems accepted for presentation at the IEEE Conference on Decision and Control (CDC) 2024 [72];

Mathematical Background 2

This Chapter introduces the fundamental mathematical concepts and definitions used throughout this work. It explores the state estimation problem from both centralized and distributed perspectives, discusses the core principles of Luenberger observers and Kalman filters, and presents techniques for fault modeling. Finally, the Observability Gramian is defined to derive a useful metric when addressing sensor selection problems. The Chapter is organized as follows: Section 2.1 presents the state estimation problem from a centralized perspective while Section 2.2 presents the state estimation problem from a distributed perspective. Section 2.3 gives details about sensor faults and, in conclusion, Section 2.4 presents the concepts of Gramian matrix and condition rank.

2.1 The State Estimation Process: a Centralized Perspective

State estimation is the process of developing a *good* estimate of the state of a process by evaluating a collection of measurements representing portions of the process state along with a mathematical representation of the overall process. The final goal is to develop the *best* possible representation of the true process state. Specifically, consider an LTI system of the form

$$x(k+1) = Ax(k) + Bu(k), \tag{2.1}$$

where $x(k) \in \mathbb{R}^{n_x}$ is the vector state, $u(k) \in \mathbb{R}^{n_u}$ the system input, $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$ are the system matrices. Suppose that the only available information about the state are provided by a set of sensors that defines the system output

$$y(k) = Cx(k), \tag{2.2}$$

where $y(k) \in \mathbb{R}^m$ is the output vector, $C \in \mathbb{R}^{m \times n_x}$ is the output matrix. One of the most important structural properties of a dynamical system is its *Observability*, i.e., the ability to estimate the initial state $x(k_0)$ from given data $(u(k), y(k))$, $\forall k \geq k_0$ and $\forall k_0 \geq 0$. This property is guaranteed if and only if the

observability matrix

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n_x-1} \end{bmatrix} \tag{2.3}$$

has full rank, i.e $rank(O) = n_x$. In general, the pair (A, C) is said to be observable if $rank(O) = n_x$. For any observer-based state estimation method, an observer is developed whose internal behavior is based on the model of the system of interest. The derived observer model is generally driven by two quantities:

- The same input signal $u(k)$ applied to the system;
- The difference between the available measurements $y(k)$ and the predicted values $\hat{y}(k|k-1)$.

By appropriately designing the observer gains, the difference between the estimated state, say $\hat{x}(k)$, and the actual system state, say $x(k)$, can be driven to zero or within a *sufficient* small set that contains zero. This allows the observer state to serve as a *good* estimate of the system state. The results presented in this work rely on two widely used state estimation techniques: the Luenberger observer and the Kalman filter.

2.1.1 Luenberger Observer

The Luenberger observer presented for deterministic linear systems in 1966 is considered as the first result of the observer-based state estimation methods [73]. The observer model is derived directly from the system model and receives as inputs, the same input of the system and its corresponding output. The observer equations are the following

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K(y(k) - \hat{y}(k)) \tag{2.4}$$

$$\hat{y}(k) = C\hat{x}(k), \tag{2.5}$$

where $\hat{x}(k) \in \mathbb{R}^{n_x}$ is the state of the observer, $\hat{y}(k) \in \mathbb{R}^m$ is the estimated output and K is the observer gain. The estimation error, defined as $e(k) \coloneqq x(k) - \hat{x}(k)$, is governed by the following equation

$$e(k+1) = (A - KC)e(k). \tag{2.6}$$

It is a common knowledge that if the pair (A, C) is observable, a gain K such that the matrix $(A - KC)$ is Shur stable can be always computed. This implies that the state of the observer converges to the true state of the system.

Remark 2.1.1. *$\hat{x}(k)$ in (2.4) and (2.4) is in some sense a prediction of $x(k)$ using information up to $k-1$.*

2.1.2 Kalman Filter

The Kalman filter is an optimal state estimator for a linear model influenced by zero-mean Gaussian noise. Consider the discrete-time linear time-invariant system

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) \\ y(k) &= Cx(k) + v(k), \end{aligned} \tag{2.7}$$

where the additional vectors $w(k) \in \mathbb{R}^{n_w}$ and $v(k) \in \mathbb{R}^{n_v}$ are the Gaussian process and system measurement noise with zero mean and covariance matrices $Q_w \in \mathbb{R}^{n_w \times n_w}$ and $Q_v \in \mathbb{R}^{n_v \times n_v}$ respectively. They are assumed statistically uncorrelated. The Kalman filter is an optimal estimator for the case where the process and measurement noises are zero-mean Gaussian noises. The filter consists in two main steps. The first step is a prediction of the state based on the previous state and on the inputs that were applied.

$$\begin{aligned} \hat{x}(k+1|k) &= A\hat{x}(k|k) + Bu(k) \\ P(k+1|k) &= AP(k|k)A^T + Q_w, \end{aligned} \tag{2.8}$$

where: $\hat{x}(k+1|k)$ is the prediction of the state $x(k+1)$ starting from all the available information at step k while $\hat{x}(k|k)$ is the state estimate at time k given the available data at time k; $P(k+1|k)$ is the covariance matrix of the prediction error $e(k+1|k) \coloneqq x(k+1) - \hat{x}(k+1|k)$ while $P(k|k)$ is the covariance matrix of the estimation error $e(k|k) \coloneqq x(k) - \hat{x}(k|k)$. This is an open-loop step and its accuracy depends completely on the quality of the model matrices A and B and on the ability to measure the inputs $u(k)$. To improve the accuracy performance the sensors measurements information is introduced using the so called innovation term

$$\upsilon(k+1) \coloneqq y(k+1) - C\hat{x}(k+1|k), \tag{2.9}$$

which is the difference between what the sensors measure ($y(k+1)$) and what the sensors are predicted to measure ($C\hat{x}(k+1|k)$). A part of this difference will be due to the noise in the sensors (the measurement noise) but the remaining discrepancy indicates that the predicted state was in error and does not properly explain the sensors observations. At this point, the second step of the Kalman filter, the update step, uses the Kalman gain

$$K(k+1) = P(k+1|k)C^T(CP(k+1|k)C^T + Q_v)^{-1} \tag{2.10}$$

to map the innovation into a correction for the predicted state, optimally tweaking the estimate based on what the sensors have observed. The resulting state estimation is

$$\begin{aligned} \hat{x}(k+1|k+1) &= A\hat{x}(k+1|k) + K(k+1)\upsilon(k+1) \\ P(k+1|k+1) &= P(k+1|k) - K(k+1)CP(k+1|k). \end{aligned} \tag{2.11}$$

The term $(CP(k+1|k)C^T + Q_v)$ is the estimated covariance of the innovation and comes from the uncertainty in the state and the measurement noise covariance. If the innovation has high uncertainty in relation to some states, this will

be reflected in the Kalman gain which will make correspondingly small adjustment to those states. The above equations constitute the classical Kalman filter which is widely used in applications from aerospace to econometrics. The filter has a number of important properties. Firstly it is recursive, the output of one iteration is the input to the next. Secondly, it is asynchronous: at a particular iteration if no sensors information is available, the prediction step is just performed with no update step. In the case that there are different sensors, each with its own C, and different sample rates, the update step is just applied using the appropriate y and C. The filter must be initialized with a reasonable value of $\hat{x}_0$ and $P(0|0)$. More precisely, x_0 has to be chosen such that $x_0 = \mathbb{E}[x_0]$, where $\mathbb{E}[\cdot]$ denotes the expected value function. The filter also requires the best possible estimates of the covariance of the process and measurement noises (Q_w and Q_v respectively).

In the literature, as an alternative to the described Kalman filter, there are also works on the use of the so called Kalman Predictor which consists in a simpler algorithm than the standard Kalman filter since it is based only on the prediction step. More formally, while the standard Kalman filter provides the estimate $\hat{x}(k|k)$ of the state $x(k)$, the Kalman predictor yields only the prediction $\hat{x}(k|k-1)$ of this state. In a mathematical point of view, the filter equations are

$$\begin{aligned} K(k) &= AP(k|k-1)C^T \left(CP(k|k-1)C^T + Q_v\right)^{-1} \\ \hat{x}(k+1|k) &= A\hat{x}(k|k-1)Bu(k) + K(k)(y(k) - C\hat{x}(k|k-1)) \\ P(k+1|k) &= (A - K(k)C)\, P(k|k-1)\, (A - K(k)C)^T + Q_w + K(k)Q_vK(k), \end{aligned} \tag{2.12}$$

where $P(k|k-1)$ is the prediction error covariance matrix related to the prediction error $e(k|k-1) = x(k) - \hat{x}(k|k-1)$. It has been proved that the Kalman predictor has the same properties of the standard Kalman filter. Obviously, in a computational point of view the predictor is less onerous than the filter since the estimation step is not performed. However due to the estimation step loss, the prediction generated by the Kalman predictor is usually worse than the estimation obtained by the Kalman filter. In conclusion it is well known that when the pair (A, C) is observable than the Kalman filter reaches a convergence state characterized by the steady-state covariance matrix

$$P = \lim_{k \to \infty} P(k+1|k). \tag{2.13}$$

Furthermore, the resulting constant matrix P is independent from the initialization value $P(0|0)$ used to initialize the filter. As a consequence, the steady state formulation of the Kalman gain given in eq. (2.10) is

$$K = PC^T(CPC^T + Q_v)^{-1}. \tag{2.14}$$

2.2 The State Estimation Process: a Distributed Perspective

Before diving within the fault tolerant estimation problem from a distributed perspective it is convenient to introduce first the distributed estimation problem.

Consider the LTI dynamical system (2.1) and suppose that the state of the system is monitored by a network of m sensors, each of which receives a partial measurement of the state at every time step. Specifically, the i-th sensor has access to a measurement of the state, given by[1]

$$y_i(k) = C_i x(k), \tag{2.15}$$

where $y_i(k) \in \mathbb{R}^{n_{y_i}}$ and $C_i \in \mathbb{R}^{n_{y_i} \times n_x}$. The aggregate output vector can be defined as follows

$$y(k) = Cx(k). \tag{2.16}$$

where $C = \left[C_1^T, \ldots, C_m^T\right]^T$ is the aggregated output matrix. These sensors are associated to nodes of an underlying communication graph $\mathcal{G} \coloneqq (\mathcal{V}, \mathcal{E}, \mathcal{A})$, which governs the information flow between the sensors. Specifically, $\mathcal{V} \coloneqq \{1, \ldots, m\}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the set of edges, and $\mathcal{A} \coloneqq [a_{ij}] \in \mathbb{R}^{m \times m}$ is the adjacency matrix whose elements satisfy $a_{ii} = 0$ and $a_{ij} > 0$ if and only if $(i, j) \in \mathcal{E}$. The matrix $\mathcal{L} \in \mathbb{R}^{m \times m}$ is the Laplacian matrix of $\mathcal{G}$. Through this work the following assumption is made

Assumption 1. *The communication graph $\mathcal{G}$ is an undirected connected graph.*

As a consequence of Assumption 1, the Laplacian matrix $\mathcal{L}$, is positive semi-definite and with exactly one zero eigenvalue. Furthermore, the Laplacian matrix of an undirected graph is symmetric. Therefore, a unitary matrix Θ always exist such that

$$blkdiag(\lambda_1 = 0, \lambda_2, \ldots, \lambda_m) = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_m \end{bmatrix} = \Theta^T \mathcal{L} \Theta, \tag{2.17}$$

where λ_i for $i = 1, \ldots, m$ are the eigenvalues of $\mathcal{L}$. For each node i, the set of its neighbors is denoted as $\mathcal{V}_i \coloneqq \{j \,|\, (i, j) \in \mathcal{E}\}$ and its cardinality is $n_i \coloneqq card(\mathcal{V}_i)$. Each node is capable of exchanging information with its neighbors and performing computational tasks. With the above ingredients it is possible to state the following *distributed estimation problem.*

Problem 2.2.1. ***(Distributed Estimation Problem)****[58]: Consider system (2.1) and a network of sensors (2.15). For each node $i = 1, \ldots, m$ construct an estimate $\hat{x}_i(k)$ of the entire system state $x(k)$ based on its respective (limited) state measurements and the information obtained from neighbors.*

Problem 2.2.1 hides several challenges. Specifically, to avoid trivial solution it is assumed that the entire network enjoys only the *joint observability* property i.e., observability of the pair (A, C). Note that the *joint observability* property does not imply local observability i.e., the pairs (A, C_i) for $i = 1, \ldots, m$ may be not observable. As a consequence, centralized solutions are not feasible in

[1] In this section systems with noiseless dynamics are considered for clarity of exposition.

this context. Solving Problem 2.2.1 requires the definition of a set of state estimate update and information exchange rules that enables the single nodes to cooperate in the reconstruction of the entire state. Formally, it is required to design a *distributed observer*.

Definition 2.2.1. ***(Distributed Observer)****[58]: A set of state estimate update and information exchange rules is called a distributed observer if* $lim_{k\to\infty}\|\hat{x}(k) - x(k)\| = 0$*, i.e., the state estimate maintained by each node asymptotically converges to the true state of the plant.*

2.3 Faults

This dissertation adopts the terminology defined by the IFAC Technical Committee: SAFEPROCESS in [74]. Specifically, a fault means an unpermitted deviation of at least one characteristic property or parameter of a system from the acceptable/usual/standard condition. It is the result of a defect in a component or subsystem which degrade the function and performance of the system. A very related term is failure which is a permanent interruption of the system's ability to perform a required function under specified operating conditions. Usually, failure means a complete breakdown of a component, whereas fault is the only deviation from normal characteristics, but a permanent fault may result in a failure. From the viewpoint of the mathematical model, faults can be modeled as external inputs or parameter deviations which change the behavior of the process. Like faults, disturbances and uncertainties can also be modeled as external inputs, and they may have similar effects on the process. However, compared to faults, disturbances and uncertainties are present even during the normal operation of the process, so they should be taken into consideration in the controller design. By contrast, faults are considered as more severe changes and their effects cannot be overcome by a fixed controller. Thus, it is necessary to detect the fault so as to prevent any serious consequences. Faults can be classified in several ways based on the components they affect, the behaviors of the faults and the way they are modeled. In this work the focus is on sensor faults.

2.3.1 Sensor Faults

Sensor faults represent the deviations between the measured and the actual value of a plant's output variable. In closed-loop systems, the measurements obtained by sensors are used to generate the control inputs. So, any fault in sensors can cause operating points far from the nominal ones, and then result in degradation in the performance of the system. Therefore, it is very important to detect these faults. Typical examples of sensor faults are listed in [75], [76]: bias, drift, performance degradation (or loss of accuracy), sensor freezing, and calibration error, as illustrated in Figure 2.1. Solid lines show the actual values whereas the dotted lines show the measured values. In this dissertation the the model based fault classification [78] is used:

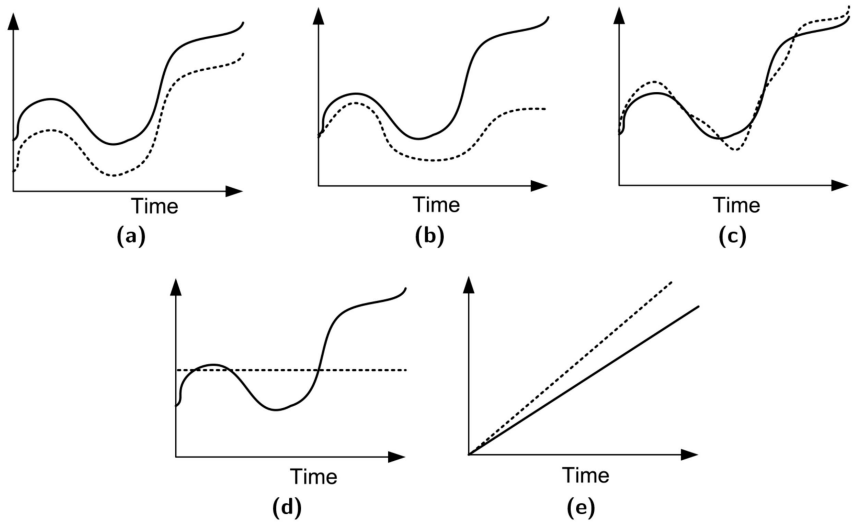

FIGURE 2.1: Common types of sensor faults: (a) bias, (b) drift, (c) performance degradation (or loss of effectiveness/accuracy), (d) sensor freezing, and (e) calibration [77].

- Additive sensor faults are modeled by an additive term that is injected directly into the output of the system. For a linear time invariant system, the corresponding output affected by a bias fault can be expressed as follows:
$$y(k) = Cx(k) + D_b b(k), \tag{2.18}$$
where $b(k) \in \mathbb{R}^{n_b}$ is the bias term and $D_b \in \mathbb{R}^{m \times n_b}$ is the bias input matrix;

- Multiplicative faults are modeled as changes in the parameter matrices C. As a consequence the faulty output of the system can be expressed as follows
$$y(k) = \Delta(\gamma(k))Cx(k), \tag{2.19}$$
where $\Delta(\gamma(k)) = diag(\gamma_1(k), \ldots, \gamma_m(k))$ is a diagonal matrix and the parameters $\gamma_i(k)$ for $i = 1, \ldots, m$ represent the effectiveness of the sensors.

Multiplicative faults on the output matrix C propagate uncertainty to both the state transition and gain matrices of the observer/filter. By considering the system output (or the innovation) as an input to the observer/filter, the gain matrix K can be treated as an input matrix. Linear Parameter-Varying (LPV) systems provide a powerful framework for modeling such time-varying systems. In the following, a general introduction to Linear Parameter-Varying (LPV) system is provided.

2.3.2 Linear Parameter Varying Systems

The foundation for Linear Parameter-Varying (LPV) systems was laid in the late 1980s (see [79], [80]) to provide a rigorous framework for gain-scheduling

control strategies. Both continuous and discrete-time LPV systems have been extensively studied in the literature. In this work, the focus is on discrete-time polytopic LPV systems (hereafter referred to as LPV systems for simplicity), which can be defined as follows [81]:

Definition 2.3.1. ***(Linear Parameter-Varying (LPV) system)****: A discrete time LPV system is a possibly time-varying linear system*

$$x(k+1) = A(\gamma_p(k))x(k) + B(\gamma_p(k))u(k), \tag{2.20}$$

where $A(\gamma_p(k))$ and $B(\gamma_p(k))$ are matrices depending linearly on a possibly time varying parameter $\gamma_p(k)$ constrained a-priori to lie in some known bounded real set[2]*.*

Definition 2.3.2. ***(Politopyc LPV system)*** *A discrete-time LPV systems is said to be polytopic when the system matrices have the following structure*

$$[A(\gamma(k)), B(\gamma(k))] = \sum_{i=1}^{n_l} \gamma_{pi}(k)[A_i, B_i], \tag{2.21}$$

where $[A_i, B_i]$, $i = 1, \ldots, n_l$ are the outmost realizations of the plants' family and the parameter vector $\gamma_p(k) = [\gamma_{p1}(k), \ldots, \gamma_{pn_l}(k)]^T$ belongs to the n_l-dimensional unit simplex

$$\mathcal{D}_{n_l} \coloneqq \left\{ \gamma_p(k) \in \mathbb{R}^{n_l} \,\middle|\, \sum_{i=1}^{n_l} \gamma_p(i) = 1,\ \gamma_{pi}(k) \geq 0 \quad i = 1, \ldots, n_l \right\}. \tag{2.22}$$

Note that the above definition does not impose any specific assumptions on the parameter behavior. This implies that, in principle, there is no assumed dependence between the current value of $\gamma_p(k)$ and its future evolution.

While this general approach offers flexibility, it may not be convenient for modeling certain classes of time-varying or nonlinear systems. In many practical applications, the parameter $\gamma_p(k)$ represents slowly varying quantities such as environmental conditions, physical parameters, or unmodeled nonlinearities. To effectively model these systems, it is necessary to introduce dependencies or constraints on the future evolution of the parameter vector, thereby refining the LPV model. The most natural way is the introduction of the so-called bounded parameter variation property:

Definition 2.3.3. ***(Bounded parameter LPV system)*** *A discrete-time polytopic LPV system is said to be subject to bounded parameter variations if $\gamma(k)$ belongs to the following possibly time-varying polytopic set*

$$\{\gamma_p(k) \in \mathcal{D}_{n_l} \mid |\gamma_{pi}(k) - \gamma_{pi}(k-1)| \leq \Delta\gamma_{pi}, \quad i = 1, \ldots, n_l\}. \tag{2.23}$$

[2]By using similar notations $\gamma_p(k)$ and $\gamma(k)$ for the LPV parameter and the multiplicative fault parameter, the idea is to emphasize the direct relationship between these two entities.

As previously discussed, LPV descriptions differ from uncertain plant representations due to the parameter which is supposed to be known (measurable or computable) at each time instant k. In order to describe all the information available at time k the information vector $x_a(k)$ is introduced

$$x_a(k) \coloneqq [x(k)^T, \gamma_p(k)^T]^T. \tag{2.24}$$

This definition is convenient because in general, at each time step k, all system decision variables (i.e. the manipulable input $u(k)$) can be chosen accordingly to the whole information vector. It is possible to distinguish between two classes of dependence, namely:

- *Time-Invariant Control Laws*: when the control law does not depend on k, i.e.
$$u(x_a(k)) = u(x(k), \gamma_p(k)); \tag{2.25}$$
- *Time-Varying Control Laws* when the control law does depend on k, i.e
$$u(x_a(k), k) = u(x(k), \gamma_p(k), k). \tag{2.26}$$

The information vector $x_a(k)$ acts as a sort of *augmented state* for the system. Starting from this consideration, model (2.20) subject to bounded parameter variations (2.23) can be rewritten as follows

$$\begin{bmatrix} x(k+1) \\ \gamma_p(k+1) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n_l} \gamma_{pi}(k) \left(A_i x(k) + B_i u(x_a(k))\right) \\ \gamma_p(k) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \Delta\gamma_p(k), \tag{2.27}$$

where $\Delta\gamma_p(k)$, the variation on $\gamma_p(k)$, can be regarded as an appropriate exogenous persistent disturbance which acts in such a way that the parameter variation constraints (2.23) are satisfied.

2.4 Condition Rank as Sensor Selection Metric

In previous sections, the observability was introduced as a structural property of a system. Note that, the observability matrix (2.3) is not useful for solving the sensor selection problem, since it only provides a binary "on/off" indication and does not contain any information about the quality of the Observability property. To consider a more appropriate observability performance criterion, the Gramian matrix

$$W = \sum_{i=1}^{n_x - 1} \left(A^T\right)^i C^T C A^i. \tag{2.28}$$

can be studied instead. Several metrics that leverage the observability Gramian as a tool for the definition of an optimization index for sensor selection tasks have been proposed in literature. In this research the focus is on a particular metric known as condition number [82]

$$\mathcal{K}(W) \coloneqq \begin{cases} \frac{\overline{\lambda}(W)}{\underline{\lambda}(W)} & \overline{\lambda}(W) > 0 \\ \infty & \overline{\lambda}(W) = 0 \ \textit{and} \ \underline{\lambda}(W) > 0 \\ 0 & W = 0, \end{cases} \tag{2.29}$$

where $\overline{\lambda}(W)$ and $\underline{\lambda}(W)$ are respectively the largest and the smallest singular values of the observability Gramian. Note that, the eigenvalues/eigenvectors of the observability Gramian W provide much information about the quality of the state reconstruction error as a function of sensor placement and selection. In particular, the minimum eigenvalue of this matrix is critical and should not be too small to avoid large reconstruction errors. The intuition that inspired preferring (within the scope of this research) the condition number over other metrics is based on the following observation: "*the condition number of the Gramian matrix measures how balanced the observability is among all modes. It grows unbounded if the system is unobservable.*" [15]. In other words, by minimizing this metric it is possible to reduce the ratio between the maximum and minimum singular values of the Gramian matrix W. Therefore, its optimizing within a sensor selection algorithm can help avoid selecting sensors whose targeted failure could severely compromise system performance, a feature that is particularly useful within a fault-tolerant state estimation scheme. Additionally, the recent study [17] suggests using the condition number as performance sensor selection criterion, since it usually leads to a smaller number of selected sensors than other criteria, such as the trace or rank of W.

Moving the focus on the optimization of the condition number, it is well-known that the condition number function is Lipschitz continuous near any positive definite matrix [82]. However, the minimization of $\mathcal{K}(\cdot)$ might be a challenging task since it cannot be performed by using classical nonlinear programming algorithms. The fundamental difficulty lies in that $\mathcal{K}(\cdot)$ is both nonconvex and not everywhere differentiable. In Chapter 3 this problem is addressed by leveraging the Simulated Annealing heuristic to reach a compromise between optimality of the determined solutions and computational complexity.

Centralized Sensor Reconciliation Design 3

This Chapter addresses the sensor reconciliation problem from a centralized perspective introducing a novel Luenberger observer-based approach. Before diving into the details of the architecture, the centralized sensor reconciliation design problem is introduced and formally defined. Specifically, the observer design problem is tackled by using LMIs. Furthermore, the fundamental question of how to select sensors for accurate and reliable state estimation is addressed as well. To overcome this issue a preliminary solution is proposed that makes use of an observability-based sensor selection optimization problem and a simulated annealing-based algorithm for its resolution. The Chapter is organized as follows: Section 3.1 introduces the centralized sensor reconciliation design problem, Section 3.2 provides the details of the architecture, Section 3.3 discusses, in a preliminary way, the sensor selection problem and presents a solution that uses a simulated annealing algorithm, Section 3.4 provides numerical results derived by the application of the architecture to the traffic flow estimation problem. In this regard the traffic model is presented and several simulated and experimental results are discussed.

3.1 Problem Statement

Lets consider a system whose dynamics is described by the following state-space representation:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_w w(k) \\ y(k) &= \Delta(\gamma(k))Cx(k) + D_b b(k), \end{aligned} \tag{3.1}$$

where: $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{m \times n_x}$, $B_w \in \mathbb{R}^{n_x \times n_w}$ and $D_b \in \mathbb{R}^{m \times n_b}$ are constant matrices. Moreover, $x(k) \in \mathbb{R}^{n_x}$ is the system state, $u(k) \in \mathbb{R}^{n_u}$ is the known input, $w(k) \in \mathbb{R}^{n_w}$ is the unknown input (i.e. a disturbance signal), $y(k) \in \mathbb{R}^m$ is the measurable *plant output* provided by physical sensors, possibly affected by bias $b(k) \in \mathbb{R}^{n_b}$ and loss of effectiveness faults modeled by the time-varying gain matrix $\Delta(\gamma(k)) = diag(\gamma_1(k), \ldots, \gamma_m(k))$. In addition, the state vector is considered to be restricted to the following set:

$$\mathcal{X} := \{x \in \mathbb{R}^{n_x} : \underline{x} \leq x \leq \overline{x}\}. \tag{3.2}$$

It must be pointed out that fault-free condition is characterized by $\Delta(\gamma(k)) = I_m$ and $b(k) = \mathbf{0}_{n_b}$ (*global healthy condition*), while in general $b(k) \neq \mathbf{0}_{n_b}$ and $\Delta(\gamma(k)) \neq I_m$ with $\gamma(k)$ that belongs to the polytope $\forall k$

$$\mathcal{Z} \subseteq \{\gamma : \mathbf{0}_m \leq \gamma \leq \mathbf{1}_m\}, \tag{3.3}$$

where the notation $\mathbf{1}_m$ and $\mathbf{0}_m$ is used to denote vectors of dimension $\mathbb{R}^m$ consisting of all ones and zeros, respectively. To ensure the system's reliability in the

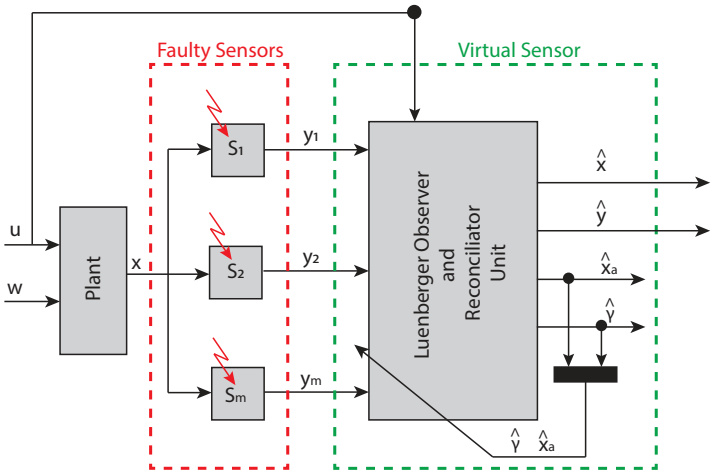

FIGURE 3.1: Fault-tolerant sensor reconciliation architecture.

event of sensor failures, it is assumed that the set of available sensors comprises redundant units. More in detail, let $\mathcal{S}^*$ be the sensor selection solution of an optimization problem (e.g. an observability based sensor selection procedure as discussed later in the Chapter). To ensure uninterrupted monitoring of a system (e.g. a traffic network) and mitigate the risks of sensor failures, it is essential to include an additional set of redundant sensors ($\mathcal{S}^R$) as backup sensors for fault tolerance. Consequently, an extended set of sensors ($\mathcal{S}^a = \mathcal{S}^* \cup \mathcal{S}^R$) with cardinality $|\mathcal{S}^a| = |\mathcal{S}^*| + |\mathcal{S}^R| = m$ must be determined to minimize the degradation of observability of the entire network link flow due to failures. In selecting the sensors, the following mild assumption should be addressed

Assumption 2. *(**Observability condition**): At each time instant k, the values assumed by the effectiveness parameters $\gamma(k)$ are such that the observability of system (3.1) is never compromised.*

Assumption 2 requires that at each time instant there are always enough "working" sensors (i.e. sensors whose associated effectiveness $\gamma_i(k) > 0$) for the state $x(k)$ being fully reconstructible. Note that it is reasonable to suppose that Assumption 2 is verified in practice when a safety degree of sensor redundancy is considered.

Remark 3.1.1. *It is important to note that considering the time varying nature of system (3.1) assuming that the observability matrix O is full rank for each time instant does not guarantee observability. A simple counterexample would*

be $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ *and* $C = I_2$. *In that case, the system is observable with a single sensor. However, the system is not observable if the two sensors fail alternatively. Formally, Assumption* 2 *requires that system (3.1) is "total" or "uniform" observable [83], [84].*

By considering that Assumption 2 holds true, the following sensor reconciliation problem needs to be solved:

Problem 3.1.1. *(**Centralized Sensor Reconciliation Design Problem**): Given system (3.1) with reference to Figure 3.1, let* $y(k)$ *be the output vector that can be affected by faults and disturbances. Develop a sensor reconciliation procedure that can hide corrupted measurements when generating state estimates based on the input signal* $u(k)$ *and the output signal* $y(k)$.

The solution of Problem 3.1.1 provides the possibility to compute an estimate $\hat{x}(k)$ of the system state $x(k)$ even with unknown input $w(k)$ and occurrence of errors on the output signal $y(k)$. In the following section, details on how to derive a centralized sensor reconciliation architecture are given.

3.2 The Virtual Sensor Design

Solving Problem 3.1.1 necessitates the development of a comprehensive fault-hiding architecture that incorporates a "*virtual sensor*" unit. The *virtual sensor* mainly consists of a *Luenberger Observer* and *Reconciliator Unit* which is in charge of:

- calculate the estimate of the system state $x(k)$, the bias term $b(k)$, and the matrix $\Delta(\gamma(k))$;
- mitigate the effects of fault occurrences on state estimates thanks to a polytopic embedding approximation of the system output and observer gain matrices.

Figure 3.1 depicts the graphical representation of the structure of the centralized sensor reconciliation architecture.

3.2.1 Luenberger Observer and Reconciliator Unit Design

The core idea of the strategy is to merge together the observer, for the estimation of the system's internal state, and the reconciliator unit, for fault detection and mitigation. To achieve this objective, a self tuning LPV observer is derived for an augmented system that takes into account the faulty parameter and their dynamics. The following augmented state, which accounts for the bias fault $b(k)$ and the multiplicative fault $\gamma(k)$, is used to design the Luenberger observer

$$x_a(k) = [x(k),\ b(k),\ \gamma_1(k),\ \ldots,\ \gamma_m(k)]^T . \tag{3.4}$$

As a consequence, the corresponding augmented model can be defined as follows

$$\begin{aligned} x_a(k+1) &= \bar{A}x_a(k) + \bar{B}u(k) + \bar{B}_w w(k) + \bar{D}_b \Delta b(k) + \bar{B}_\gamma \Delta\gamma(k) \\ \bar{y}(k) &= \bar{C}_{\gamma(k),x(k)} x_a(k), \end{aligned} \tag{3.5}$$

where

$$\begin{aligned} \Delta b(k) &= b(k+1) - b(k) \\ \Delta\gamma(k+1) &= \gamma(k+1) - \gamma(k), \\ \bar{A} = \begin{bmatrix} A & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, \bar{B} &= \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix}, \\ \bar{B}_w = \begin{bmatrix} B_w \\ 0 \\ 0 \end{bmatrix}, \bar{D}_b = \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix}, \bar{B}_\gamma &= \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}, \\ \bar{C}_{\gamma(k),x(k)} = \begin{bmatrix} \Delta(\gamma(k))C & D_b & 0 \\ 0 & D_b & diag(Cx(k)) \end{bmatrix}&. \end{aligned} \tag{3.6}$$

Note that the fault signal $\gamma(k)$, the bias term $b(k)$ and, thus, their corresponding increments $\Delta b(k)$, and $\Delta\gamma(k)$ are assumed to be bounded in the sense of the l_2-norm.

Let $\hat{\gamma}(k)$ and $\hat{x}(k)$ be the estimates of the multiplicative fault $\gamma(k)$ and the system state $x(k)$ at each time k, respectively. Then it is possible to define a state observer for the augmented system (3.5) as

$$\begin{cases} \hat{x}_a(k+1) = \bar{A}\hat{x}_a(k) + \bar{B}u(k) + K_{\hat{\gamma}(k),\hat{x}(k)}(\bar{y}(k) - \hat{y}(k)) \\ \hat{y}(k) = \bar{C}_{\hat{\gamma}(k),\hat{x}(k)}\hat{x}_a(k). \end{cases} \tag{3.7}$$

Defining the error of the state estimate as $e(k) \coloneqq x_a(k) - \hat{x}_a(k)$, the following relation emerges for the error dynamics

$$\begin{aligned} e(k+1) = (\bar{A} - K_{\hat{\gamma}(k),\hat{x}(k)}\bar{C}_{\hat{\gamma}(k),\hat{x}(k)})e(k) + \bar{B}_w w(k) + \bar{D}_b \Delta b(k) + \\ + \bar{B}_\gamma \Delta\gamma(k) + \underbrace{K_{\hat{\gamma}(k),\hat{x}(k)}(\bar{C}_{\hat{\gamma}(k),\hat{x}(k)} - \bar{C}_{\gamma(k),x(k)})x_a(k)}_{d(k)}, \end{aligned} \tag{3.8}$$

where $d(k)$ might be considered as a disturbance term. Specifically, the magnitude of $d(k)$ is not affected by its historical values ($d(k-1)$, $d(k-2)$ ...) but it is directly related to the quality of the estimated state and effectiveness values. In case of perfect estimation $\bar{C}_{\hat{\gamma}(k),\hat{x}(k)} = \bar{C}_{\gamma(k),x(k)}$ resulting in a zero added disturbance error. If the additive terms in (3.8) are bounded, i.e. $\Delta b(k)$ and $\Delta\gamma(k)$ remain bounded and the $x \in \mathcal{X}$, then the boundedness of the estimation error depends on the evolution of the following simplified time varying system

$$e(k+1) = (\bar{A} - K_{\hat{\gamma}(k),\hat{x}(k)}\bar{C}_{\hat{\gamma}(k),\hat{x}(k)})e(k) + \bar{B}_w w(k). \tag{3.9}$$

Moved by the above consideration, the design procedure considers the hypothesis that the system (3.8) is characterized by uncertainty with respect to γ and x, and attempts to determine an LPV gain that can be tuned online using

only the estimates $\hat{\gamma}(k)$ and $\hat{x}(k)$ of the true $\gamma(k)$ and $x(k)$. In this context, $\mathbb{R}$ denotes the set of real numbers. It is possible to define a generic ball in a Euclidean n_x-space $\mathbb{R}^{n_x}$ as $\mathcal{B}_\epsilon := \{x \in \mathbb{R}^{n_x} : |x|_2 \leq \epsilon\}$. Furthermore, the following polytopic embedding approximation is assumed for the matrix $\bar{C}_{\hat{\gamma}(k),\hat{x}(k)}$ given by

$$\bar{C}_\rho = \sum_{i=1}^{n_l} \rho_i(\hat{\gamma}, \hat{x}) \bar{C}_i^v, \tag{3.10}$$

where matrices $\bar{C}_i^v$, $i = 1, \ldots, n_l$ are the polytope vertices and $\rho_i : \mathcal{I} \times \mathcal{X} \sim \mathcal{B}_\epsilon \to \mathbb{R}$ is a continuous function of $\hat{\gamma}$ and $\hat{x}$. Moreover, by introducing the following auxiliary normalized functions:

$$\kappa_1(\hat{\gamma})^{(j_1)} = \frac{\hat{\gamma}^{(j_1)}(k) - \underline{\hat{\gamma}}^{(j_1)}}{\overline{\hat{\gamma}}^{(j_1)} - \underline{\hat{\gamma}}^{(j_1)}}, \qquad \kappa_2(\hat{x})^{(j_2)} = \frac{\hat{x}^{(j_2)}(k) - \underline{\hat{x}}^{(j_2)}}{\overline{\hat{x}}^{(j_2)} - \underline{\hat{x}}^{(j_2)}}, \tag{3.11}$$

$$j_1 = 1, \ldots, m; \qquad j_2 = 1, \ldots, n_x;$$

the scheduling parameter $\rho_i(\hat{\gamma}, \hat{x})$ can be defined as

$$\begin{cases} \rho_1(\hat{\gamma}, \hat{x}) & = \Pi_{j_1,j_2} \kappa_1(\hat{\gamma})^{(j_1)} \kappa_2(\hat{x})^{(j_2)} \\ & \vdots \\ \rho_{n_l}(\hat{\gamma}, \hat{x}) & = \Pi_{j_1,j_2} (1 - \kappa_1(\hat{\gamma})^{(j_1)})(1 - \kappa_2(\hat{x})^{(j_2)}). \end{cases} \tag{3.12}$$

Note that $\sum_{i=1}^{n_l} \rho_i(\hat{\gamma}, \hat{x}) = 1$ and $0 \leq \rho_i(\hat{\gamma}, \hat{x}) \leq 1$.

Then one attempts to design a discrete-time self-tuning LPV observer whose matrix gain $K_{\hat{\gamma},\hat{x}(k)}$ is such that:

- stabilizes the error dynamics (3.8) despite the occurrence of system faults and disturbances;
- it can be tuned in real-time by exploiting an estimate $\hat{\gamma}(k)$, $\hat{x}(k)$ of the true $\gamma(k)$, $x(k)$.

Subsequently, the goal is to design an LPV observer gain (by solving a suitable LMI optimization problem) as

$$K_\rho = \sum_{i=1}^{n_l} \rho_i(\hat{\gamma}, \hat{x}) K_i, \tag{3.13}$$

where the gains K_i, $i = 1, \ldots, n_l$ are properly designed so that the error dynamics

$$e(k+1) = (\bar{A} - K_\rho \bar{C}_\rho) e(k) + B_{\bar{w}} \bar{w}(k), \tag{3.14}$$

with

$$B_{\bar{w}} \coloneqq \begin{bmatrix} \bar{B}_w & \bar{D}_b & \bar{B}_\gamma & I \end{bmatrix}, \tag{3.15}$$

$$\bar{w}(k) \coloneqq \begin{bmatrix} w(k)^T & \Delta b(k)^T & \Delta\gamma(k)^T & d(k)^T \end{bmatrix}^T, \tag{3.16}$$

converges to zero for $k \to \infty$. In this respect, by defining the performance output as

$$z_e(k) = \bar{C}_\rho e(k) \tag{3.17}$$

the observer gains can be obtained by minimizing the following $\mathcal{H}_\infty$ performance index

$$J := \sup_{\bar{w}\in l_2,\ \rho:\mathcal{I}\times\mathcal{X}\sim\mathcal{B}_\epsilon\to\mathbb{R}} \frac{\|z_e\|_2}{\|\bar{w}\|_2}. \tag{3.18}$$

In this context, Proposition 3.2.1 provides a convex optimization strategy to solve the aforementioned observer gain design problem.

Proposition 3.2.1. *With reference to our case of study, the observer gain design problem can be solved if the following LMI optimization problem in the unknown matrices $P = P^T > 0$ and Y_i, $i = 1, \ldots, n_l$*

$$\min_{P,\ Y_i,\ \alpha} \alpha^2 \tag{3.19}$$

$$\begin{bmatrix} P & P\bar{A} - Y_i\bar{C}_i^v & P\bar{B}_{\bar{w}} & 0 \\ * & P & 0 & \left(\bar{C}_i^v\right)^T \\ * & * & \alpha^2 I & 0 \\ * & * & * & \alpha^2 I \end{bmatrix} > 0, \quad i = 1, \ldots, n_l \tag{3.20}$$

admits a solution. Then, the observer gains to be used in (3.13) are given by $K_i = P^{-1} Y_i$, $i = 1, \ldots, n_l$.

Proof. Consider the following discrete-time state-space representation

$$\begin{aligned} e(k+1) &= (\bar{A} - K_\rho \bar{C}_\rho)e(k) + (B_{\bar{w}} - K_\rho D_d)\bar{w}(k) \\ z_e(k) &= \bar{C}_\rho e(k) + D_d \bar{w}(k), \end{aligned} \tag{3.21}$$

and the dissipation inequality

$$V(e(k+1)) - V(e(k)) \leq s(\bar{w}(k), z_e(k)), \quad \forall k \in \mathbb{N}, \tag{3.22}$$

where:

- $V : \mathbb{R}^{(n_x+n_b+m)} \to \mathbb{R}$ is a storage function that generalizes the energy function for a dissipative system;
- $s : \mathbb{R}^{(n_w+n_x+n_b+m)} \times \mathbb{R}^{2m} \to \mathbb{R}$ is a supply function representing the rate at which the system absorbs energy.

Starting from the definition of $\mathcal{H}_\infty$ norm it is possible to write

$$\frac{\|z_e(k)\|_2}{\|\bar{w}(k)\|_2} < \alpha \to \|z_e(k)\|_2 < \alpha \|\bar{w}(k)\|_2 . \tag{3.23}$$

The supply function $s(\bar{w}(k), z_e(k))$ can be defined as

$$s(\bar{w}(k), z_e(k)) = \alpha^2 \bar{w}^T(k)\bar{w}(k) - z_e(k)^T z_e(k), \tag{3.24}$$

that, by considering the state-space representation (3.21) and eqs. (3.10)-(3.15), can be rewritten as

$$s(\bar{w}(k), \bar{C}_i^v e(k) + \bar{B}_{\bar{w}}\bar{w}(k)) = \alpha^2 \bar{w}^T(k)\bar{w}(k) - \\ - \left[\bar{C}_i^v e(k) + D_d \bar{w}(k)\right]^T \left[\bar{C}_i^v e(k) + D_d \bar{w}(k)\right], \ \forall i. \quad (3.25)$$

The storage function related to the supply function $s(\bar{w}(k), z_e(k))$ can be defined as $V(e(k)) = e^T(k)Pe(k)$, $P = P^T > 0$. It can be viewed as a Lyapunov candidate function. Consequently, it is possible to compute

$$\begin{aligned} \Delta V(e(k)) &= V(e(k+1)) - V(e(k)) \\ &= e(k+1)^T P e(k+1) - e(k)^T P e(k). \end{aligned} \quad (3.26)$$

Then, by replacing (3.21) in (3.26), it is possible to obtain

$$\Delta V(e(k)) = \left[M_1 e(k) + M_2 \bar{w}(k)\right]^T P \left[M_1 e(k) + M_2 \bar{w}(k)\right] - e(k)^T P e(k), \ \forall i, \quad (3.27)$$

where $M_1 = (\bar{A} - K_i \bar{C}_i^v)$ and $M_2 = (\bar{B}_{\bar{w}} - K_i D_d)$ are matrices whose dependency by the index i as been omitted for simplicity. By substituting equations (3.27) and (3.25) in (3.22), the following inequality is obtained

$$\left[M_1 e(k) + M_2 \bar{w}(k)\right]^T P \left[M_1 e(k) + M_2 \bar{w}(k)\right] + \\ + \left[\bar{C}_i^v e(k) + D_d \bar{w}(k)\right]^T \left[\bar{C}_i^v e(k) + D_d \bar{w}(k)\right] - \\ - e(k)^T P e(k) - \alpha^2 \bar{w}^T(k)\bar{w}(k) < 0, \quad \forall i. \quad (3.28)$$

Equation (3.28) can be also compacted in the following form

$$\begin{bmatrix} e(k) \\ \bar{w}(k) \end{bmatrix}^T \underbrace{\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}}_{M} \begin{bmatrix} e(k) \\ \bar{w}(k) \end{bmatrix} < 0, \quad \forall i, \quad (3.29)$$

where:

- $M_{11} = (\bar{A} - K_i \bar{C}_i^v)^T P (\bar{A} - K_i \bar{C}_i^v) - P + \left(\bar{C}_i^v\right)^T \bar{C}_i^v$;
- $M_{12} = (\bar{A} - K_i \bar{C}_i^v)^T P (\bar{B}_{\bar{w}} - K_i D_d) + \left(\bar{C}_i^v\right)^T D_d$;
- $M_{21} = M_{12}^T$;
- $M_{22} = (\bar{B}_{\bar{w}} - K_i D_d)^T P (\bar{B}_{\bar{w}} - K_i D_d) + D_d^T D_d - \alpha^2 I$.

From inequality (3.29) it is evident that $M < 0$. Moreover the matrix M can be decomposed as the sum of two matrices

$$M = \begin{bmatrix} M_1^T P M_1 - P & M_1^T P M_2 \\ * & M_2^T P M_2 - \alpha^2 I \end{bmatrix} + \begin{bmatrix} \left(\bar{C}_i^v\right)^T \\ D_d^T \end{bmatrix} \begin{bmatrix} \bar{C}_i^v & D_d \end{bmatrix} < 0 \quad (3.30)$$

Then, by applying the Schur complement to (3.30) it is possible to obtain

$$\begin{bmatrix} M_1^T P M_1 - P & M_1^T P M_2 & \left(\bar{C}_i^v\right)^T \\ * & M_2^T P M_2 - \alpha^2 I & D_d^T \\ * & * & -I \end{bmatrix} < 0 \tag{3.31}$$

that, by choosing D_d equal to the zero matrix of appropriate dimensions and by introducing $Y_i = K_i P$, is equivalent to (3.20) on the basis of the Discrete-Time Bounded Real Lemma [85], [86]. ■

Remark 3.2.1. *The stability of the overall architecture was here established using a common Lyapunov function for all polytope vertices. This approach although effective might be conservative. Alternatively, stability could be achieved by defining specific operating conditions, each with its own Lyapunov function. Faults could then be mitigated by switching between these conditions. Stability in this switching scheme can be guaranteed by ensuring that the system evolve on the selected mode for a "long enough" time. Such a time is commonly referred in literature as "dwell time". A similar approach is explored in a distributed context to prove the stability of the Distributed Sensor Reconciliation Architecture proposed in Chapter 4.*

3.3 Observability Based Sensor Selection Procedure

To delve deeper into the closely related topic of sensor selection, the process of selecting an appropriate set of sensors that guarantees observability while ensuring a desired level of redundancy is going to be explored as well. In this section a preliminary result about the selection of sensors is discussed. In particular the sensor selection problem is formulated as a constrained optimization problem and solved using the simulated annealing heuristic.

3.3.1 Sensor Selection Problem

Let $\mathcal{S}$ denote the set of all n_x possible sensor positions, where each element $s_i \in \mathcal{S}$ represents the i-th sensor position. The task is to create a process for selecting sensors that allows to choose a minimal sub-group of r sensors from m sensor positions. The aim is to optimize the performance criterion (2.29) whilst ensuring that the observability of the road network system is maintained.

The goal of the sensor selection procedure is, given a generic LTI system, to select a subset $\mathcal{J} \subseteq \mathcal{S}$ of r sensors from the set of m potential sensors to optimize a measure of observability. Therefore, the following mixed-integer constrained optimization problem can be used to describe the solution to the observability-based sensor selection problem in terms of the observability Gramian [87]

$$\begin{aligned} \min_{h} \quad & \mathcal{K}[W(h)] \\ s.t. \quad & \sum_{i=1}^{n} h_i = r \\ & h_i \in \{0,1\}, \ i = 1 \ldots, m, \end{aligned} \tag{3.32}$$

where $\mathcal{K}[\cdot]$ is a scalar measure of the observability Gramian (i.e. the condition number) [87] and the budget constraint in (3.32) dictates that exactly r sensors must be selected. In addition, it should be emphasized that for the problem of sensor selection considered here:

- the following output equation is considered with respect to the model (3.1) when the system evolve in a healthy condition

$$y^*(k) = C(\mathcal{J})x(k), \tag{3.33}$$

  where $C(\mathcal{J}) \in \mathbb{R}^{r \times n_x}$ indicates the components of the state selected as measurement output. It has r rows extracted from the $n_x \times n_x$ identity matrix in accordance to the index set $\mathcal{J} = \{s_1, \ldots, s_r\}$;

- the observability Gramian (see Chapter 2 eq. (2.28) for a formal definition) is reformulated as

$$W(h) = \sum_{i=1}^{n_x} h_i \sum_{j=1}^{n_x - 1} (A^T)^j C(\mathcal{J})^T C(\mathcal{J}) A^j, \tag{3.34}$$

  where h_i with $i = 1, \ldots, n_x$, are binary "activation variables" whose meaning can be summarized by the function

$$f(\mathcal{J}) = \begin{cases} h_i = 1, \text{sensor } s_i \text{ is selected} \\ h_i = 0, \text{sensor } s_i \text{ is NOT selected.} \end{cases} \tag{3.35}$$

It is important to remember that the optimization problem (3.32) is a non-linear non-convex integer programming problem. It involves finding a finite number of sensor locations that satisfy predefined constraints. Exhaustive search (also known as brute force strategy), which methodically lists all potential candidate solutions and determines whether each one satisfies the problem's statement, is one approach that could be used to solve this problem. However, the exhaustive search is NP-hard [88], and the evaluation of each combination becomes unpractical for large networks [89]. To solve (3.32) and obtain an adequate set of sensors $\mathcal{S}^* \subseteq S$ with $|\mathcal{S}^*| = r$, suboptimal strategies can be used, such as the heuristic method proposed in this work. It is important to recall that the exhaustive search approach allows the discovery of a global mathematical optimum $\mathcal{S}^{opt}$. In contrast, heuristic methods are not always guaranteed to reach the optimal selection $\mathcal{S}^* = \mathcal{S}^{opt}$.

Since the optimization problem (3.32) is a nonlinear and non-convex mixed-integer programming problem [17], a global optimization solver must be used. Among many others, the ability of the Simulated Annealing (SA) and Genetic Algorithm (GA) search heuristics to find a suitable solution is investigated here. These solutions are then compared to the solution of the brute force enumeration approach. In particular, the focus of this work is on the SA algorithm. However, in the final section, the results of all three methods: SA, GA and brute force; are compared.

3.3.2 Simulated Annealing Techniques

The focus within the scope of this research is on the Simulated Annealing (SA) algorithm. This heuristic enable finding a trade off between optimality of and computational burden of the algorithm. Specifically, SA is a local search method that, in particular when working with discrete search spaces, seeks to find a global optimum in a large search space. The name is a reference to the metallurgical process of annealing, in which materials are heated and cooled in a carefully controlled manner. SA attempts to avoid being trapped in a local optimum by occasionally accepting to increase the cost by moving to a neighboring position [90], unlike traditional local search algorithms (e.g., the descendant algorithm) whose disadvantage is that the local minimum reached may be far from any global minimum. When applied to a discrete optimization problem, the values of two solutions (the current solution and a newly selected solution) are compared at each iteration of a SA procedure. Improving solutions are always accepted, while some of the non-improving (inferior) solutions are accepted in the hope of getting out of the local optimum in the search for a global optimum. The probability that non-improving solutions are accepted depends on a *temperature parameter* that usually decreases with each iteration of the procedure.

The following statements are introduced for the formal description of the SA algorithm:

- the *solution space* Ξ (i.e., the set of all possible solutions):

$$\Xi = \{\mathcal{J} \subseteq \mathcal{S} : rank(W(h)) = n_x;\ \ h = f(\mathcal{J})\}\,; \tag{3.36}$$

- the objective function $\mathcal{K}[W(h)]$ defined on the solution space.

The objective consists in finding a global minimum, $\mathcal{S}^*$, i.e., $\mathcal{S}^* \subseteq \Xi$, such that $\mathcal{K}[W(h^*)] \leq \mathcal{K}[W(h)]$ for all $\mathcal{J} \in \Xi$. Define $f'(\mathcal{J})$ as the neighborhood set for $\mathcal{J}$. Therefore, for each solution $\mathcal{J} \in \Xi$ there are neighboring solution $\mathcal{J}' \in f'(\mathcal{J})$ that can be reached in a single iteration of a local search algorithm. The SA algorithm starts with an initial solution $\mathcal{J} \in \Xi$. A neighboring candidate solution $\mathcal{J}' \in f'(\mathcal{J})$ is generated either randomly or according to a given rule. SA follows the Metropolis acceptance criterion [90], which shows how a thermodynamic system changes from the current solution (state) $\mathcal{J} \in \Xi$ to a candidate solution $\mathcal{J}' \in f'(\mathcal{J})$, where the energy content is minimized. The candidate solution $\mathcal{J}'$ is accepted as the current solution based on the following acceptance probability:

$$f_p\left(\mathcal{J}'\right) = \begin{cases} exp(-(\delta/t_i)) & if\ \delta > 0 \\ 1 & otherwise, \end{cases} \tag{3.37}$$

where $\delta = \mathcal{K}[W(h')] - \mathcal{K}[W(h)]$ and t_i is the temperature parameter at iteration i, such that

$$t_i := \left\{ t_i > 0, \forall i, \lim_{i \to \infty} t_i = 0 \right\}. \tag{3.38}$$

SA PSEUDO-CODE
INPUT: r
OUTPUT: $\mathcal{S}^*$
OBJECTIVE FUNCTION TO BE MINIMIZED: $\mathcal{K}[W(h)]$

Select an initial solution $\mathcal{J} \in \Xi$ and define $h = f(\mathcal{J})$
Select the temperature change counter $i = 0$
Select a temperature cooling schedule, t_i
Select an initial temperature $T = t_0 \geq 0$
Select a repetition schedule, c_{max}, that defines the number of iterations executed at each temperature, t_i
do
    Set repetition counter $c_{rep} = 0$
    do
        Generate a solution $\mathcal{J}' \in f'(\mathcal{J})$
        Define $h' = f(\mathcal{J}')$
        Compute $\delta = \mathcal{K}[W(h')] - \mathcal{K}[W(h)]$
        if $\delta \leq 0$ **then**
            $\mathcal{J} = \mathcal{J}'$
            $h = h'$
        else
            $\mathcal{J} = \mathcal{J}'$
            $h = h'$
            with probability $exp(-\delta/t_i)$
        end if
        $c_{rep} \leftarrow c_{rep} + 1$
    while not $c_{rep} == c_{max}$
    $i \leftarrow i + 1$
    $T \leftarrow t_i$
while not $|\mathcal{J}| == r$
$\mathcal{S}^* = \mathcal{J}$

Remark 3.3.1. *It is important to emphasize that the described procedure only allows optimal sensor placement, but not the definition of a sensor configuration that enable fault-tolerant characteristics. To this end, it is necessary to develop an ad hoc procedure that, by using a redundant set of sensors, is able to mitigate the effects of the loss of sensor effectiveness and the occurrence of faults. In this regard, it is possible to extend the sensor subset $\mathcal{S}^*$ provided by the SA algorithm by considering an alternative feasible sensor subset that can be used as backup sensors in case of failures. In this respect, the selection of backup sensors can be done by analyzing the network configuration, looking for backup sensors that guarantee the observability of the system. Another possible solution is to consider the fault combinations among the most fault-prone sensors and then try to find the location of the redundant sensors by searching for them in alternative solutions provided by SA or other any sensor selection methods (e.g. GA) [91]*

3.4 Numerical Examples

To evaluate the efficacy of the centralized sensor reconciliation architecture, simulations and experiments were conducted within a simulated urban network. This section outlines the traffic model employed for numerical tests and presents the results of numerical simulations. The observer used to derived the virtual sensor is constructed on the base of the sensors selected using the simulated annealing algorithm. Additionally, an experimental validation is conducted using a realistic simulator to further assess the architecture's performance.

3.4.1 Model Definition

A traffic network can be thought of as a WSN in which a collection of sensors (such as motion sensors, cameras, radar, etc.) is responsible for reporting data on traffic volumes in each segment of the monitored road network. Moreover, the dynamic flow of vehicles within the urban network can be modeled as a compartmental model as proposed in [92]. The structure of the compartmental model is not complex from a conceptual point of view: the system is divided into homogeneous compartments and the flow between these compartments is traced. To keep the discussion more general, reference can be made to the Figure 3.2, which shows a simple example of a traffic compartmental system. In this Figure, the compartments (road sectors) are represented by boxes and the flows by arrows. Also, u_i is the number of vehicles entering the i-th road segment, while y_i is the number of vehicles exiting the i-th road segment ($i = 1, \ldots, 5$). Note also that some of the vehicles leaving the i-th road sector may represent the input of the j-th road sector - e.g., y_3 becomes an input of R_4 - ($i = 1, \ldots, 5$, $j = 1, \ldots, 5$, $j \neq i$).

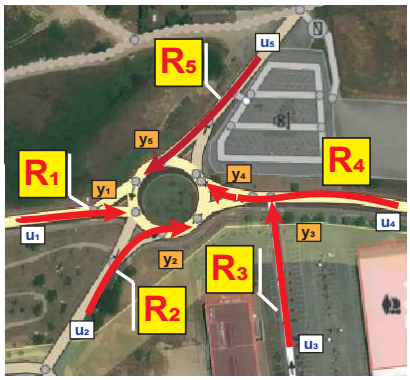

FIGURE 3.2: An example of traffic compartmental system in a real world scenario.

The compartmental system can be conceptualized also as a direct graph in which the compartments are the vertices and the flows between them are the arcs. This similarity is also evident in Figure 3.2, which shows the structural aspect of the system. Referring to the road network model definition, the number of vehicles within the i-th compartment can be denoted as $x_i(k)$. Furthermore, it

is assumed that the following condition is satisfied for each compartment:

$$x_i(k) \geq 0, \forall k \geq 0. \tag{3.39}$$

The flow from the i-th to the j-th compartment is assumed to be proportional to $x_i(k)$ according to a non-negative constant splitting ratio a_{ji}

$$f_{ji} = a_{ji}x_i(k), \quad j \neq i, \quad a_{ji} \geq 0. \tag{3.40}$$

More specifically, when several roads (e.g. R_1, R_2, R_4 and R_5 in Figure 3.2) intersect, not all vehicles in a given section R_i are forced to move towards a single road section, say R_j. As a consequence, the splitting ratio a_{ji} denotes the fraction of vehicles that moves from section R_i towards section R_j. In addition, it is important to note that the model does not take into account certain compartments, which are conventionally referred to as the singular "virtual" compartment 0. This compartment is known as the "external world", and the flow from it to the compartment i serves as the system input. Conversely, the flow from compartment i to the external world is modeled as $f_{0i} = a_{0i}x_i(k)$.

Given a region of interest $\mathcal{T}$, the set of all road sectors contained therein can be defined as $\tilde{\mathcal{R}} = \{R_1, \ldots, R_{n_r}\}$. It is important to note that not all road segments belonging to $\mathcal{T}$ are of equal importance. For example, the online vector map databases (e.g. OpenStreetMap - OSM) [1] provide a comprehensive ranking of roads grouped into different *importance categories*. In the following, seven categories of roads are considered, with the least important ones rated $c_i = 1$ and the most important ones rated $c_i = 7$. Furthermore, road sections with low relevance for traffic monitoring applications (e.g. cul-de-sacs) can be excluded from the initial set $\tilde{\mathcal{R}}$. This reduces the initial set $\tilde{\mathcal{R}}$ to the set of main roads $\mathcal{R} = \{R_1, \ldots, R_{n_x}\} \in \tilde{\mathcal{R}},\ n_x \leq n_r$. For each road sector $R_i \in \mathcal{R}$:

1. the OSM online vector maps provide the importance category;
2. $\mathcal{P}$ represents the set of road sectors connected to the outside world;
3. the matrix G is defined, whose elements are given as follows:

$$g_{ji} = \begin{cases} 1 & if \quad R_i \to R_j \\ 0 & otherwise, \end{cases} \tag{3.41}$$

   where $R_i \to R_j$ means that vehicles can travel from road R_i to road R_j. Note that G is not symmetric in general because of the existence of possible one-way streets.

Then the compartmental road network can be modeled as an LTI system [17], where:

- the state $x(k) \in \mathbb{R}^{n_x}$ is defined such that the quantity $x_i(k)$ indicates the actual traffic flow (number of vehicles per time unit) on the road sector R_i;

[1] OpenStreetMap, https://www.openstreetmap.org

- the splitting ratios defined according to the importance rank of the roads are the elements of the state matrix $A \in \mathbb{R}^{n_x \times n_x}$. In particular, the off-diagonal elements a_{ji} with $i \neq j$ can be defined as follows

$$a_{ji} = \begin{cases} \frac{c_j}{\sum_{l:R_i \to R_l} c_l + c_i} & if\ R_i \to R_j\ \&\ R_i \notin \mathcal{P} \\ \frac{c_j}{\sum_{l:R_i \to R_l} c_l + 2c_i} & if\ R_i \to R_j\ \&\ R_i \in \mathcal{P} \\ 0 & otherwise, \end{cases} \tag{3.42}$$

  while the diagonal components a_{ii} are computed as

$$a_{ii} = 1 - \left(a_{0i} + \sum_{j \neq i} a_{ji} \right) \tag{3.43}$$

  where

$$a_{0i} = \frac{c_0}{\sum_{l:R_i \to R_l} c_l + 2c_i} \tag{3.44}$$

  is the splitting ratio of the traffic flow from the i-th road sector R_i to the external world.

- The flow of traffic from the external world into the road sector R_i is defined as the i-th entry of the system input. As a consequence, the input matrix is a diagonal matrix $B \in \mathbb{R}^{n_x}$ that as a 1 on the i-th diagonal term if R_i is connected with the external world;

- $y(k) \in \mathbb{R}^{n_x}$ is such that $y_i(k)$ denotes the number of vehicles leaving the road sector R_i.

The relation expressed in eq. (3.42) implies that the flow from R_i to R_j is proportional to the importance of R_j, normalized w.r.t. the importance of all other roads connected to R_i (and R_i itself). In addition, if $R_i \in \mathcal{P}$, the flow towards the external world, whose importance score is by convention equal to that of R_i itself, should be considered as well.

3.4.2 Simulation Results

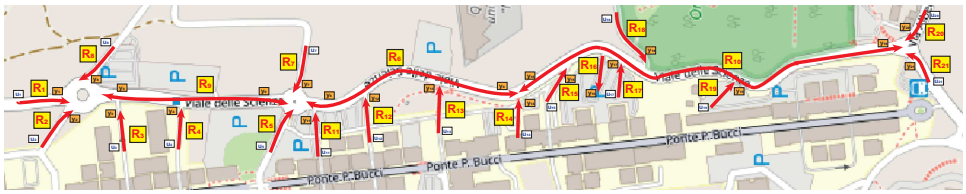

FIGURE 3.3: Example of a traffic compartmental system. Road schematic around the University of Calabria. R_i denote 21 potential sensor locations, u_i and y_i the vehicles entering and leaving the i-th link, respectively.

The objective of this section is to illustrate the applicability of the proposed estimation procedure using a traffic network composed of $\mathcal{R} = \{R_1, \ldots, R_{21}\}$

road segments (see Figure 3.3). Simulative tests have been performed on a workstation equipped with 2 × Intel® Xeon® Gold 5218 Processor 16-Core 2.3GHz and 12 × 32GB PC4-23400 2933MHz DDR4 RAM. The first step was to determine the road network model. In this context, the system matrices (3.45) and (3.46) were calculated. Note that the rows of the input matrix that refer to the road sectors that are not connected to the *external world* (i.e., R_6, R_9, R_{10}, R_{16}) are null.

$$A = \begin{pmatrix}
0.13 & 0.33 & 0 & 0 & 0 & 0 & 0 & 0.27 & 0.27 & 0 & 0 \\
0.30 & 0.30 & 0 & 0 & 0 & 0 & 0 & 0.23 & 0.17 & 0 & 0 \\
0 & 0 & 0.85 & 0 & 0 & 0 & 0 & 0 & 0.15 & 0 & 0 \\
0 & 0 & 0 & 0.85 & 0 & 0 & 0 & 0 & 0.15 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.23 & 0.2 & 0.23 & 0 & 0.14 & 0 & 0.2 \\
0 & 0 & 0 & 0 & 0.14 & 0.09 & 0.17 & 0 & 0.01 & 0.14 & 0.09 \\
0 & 0 & 0 & 0 & 0.19 & 0.05 & 0.45 & 0 & 0.01 & 0 & 0.30 \\
0.42 & 0.21 & 0 & 0 & 0 & 0 & 0 & 0.36 & 0.01 & 0 & 0 \\
0.15 & 0.16 & 0.15 & 0.15 & 0.10 & 0.01 & 0.01 & 0.14 & 0.01 & 0 & 0.12 \\
0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0.27 & 0 \\
0 & 0 & 0 & 0 & 0.34 & 0.15 & 0.14 & 0 & 0.08 & 0 & 0.29 \\
0 & 0 & 0 & 0 & 0 & 0.10 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.20 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.10 & 0 & 0 & 0 & 0.06 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.05 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.10 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.10 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.08 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.09 & 0
\end{pmatrix}$$

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.10 & 0.20 & 0.06 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.10 & 0.05 & 0 & 0.11 & 0.1 & 0.1 & 0.1 & 0.07 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.90 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.80 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.84 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.95 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1.00 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.89 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.90 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.90 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.90 & 0.02 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.91
\end{pmatrix} \tag{3.45}$$

$$B = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \tag{3.46}$$

The road network system observability is then ensured by selecting a minimal subset of r sensors from a collection of m potential sensor distributions. In this

respect, a solution to the sensor selection problem was achieved by appropriately adapting the built-in MATLAB functions `simulannealbnd` for integer/discrete optimization. This is achieved by truncating the generated floating-point numbers within meaningful integral windows [93]. The problem with $n_x = 21$ is still tractable and it is possible to find the minimal subset $\mathcal{S}_{sa}^*$ of $\mathcal{S}$ which guarantees the complete observability of the system, that is

$$rank(O(A, C(\mathcal{S}_{sa}^*))) \quad = \quad 21. \tag{3.47}$$

In particular, the following results were obtained as a solution to the sensor selection problem

- Computation time: 94.5[sec];
- Cardinality $|S_{sa}^*| = 6$;
- $\mathcal{K}(W) = 4.1102 \times 10^3$;
- $\mathcal{S}_{sa}^* = [s_3,\ s_{13},\ s_{15},\ s_{17},\ s_{19},\ s_{20}]$.

Further tests were carried out to investigate whether the solution obtained through the SA process represents the minimum sensor set required for the system to be observable. The basic idea is to find possible alternative sensor configurations useful to build a redundant sensor set. In this regard, two types of procedures were considered: 1) GA sensor selection procedure as in [17]; 2) brute force search (exhaustive). Using the GA procedure, following results have been obtained:

- Computation time: 89.6[sec];
- Cardinality $\left|S_{ga}^*\right| = 7$;
- $\mathcal{K}(W) = 7.6840 \times 10^4$;
- $\mathcal{S}_{ga}^* = [s_3,\ s_5,\ s_{12},\ s_{15},\ s_{17},\ s_{19},\ s_{20}]$.

On the other hand, the exhaustive search provides the results reported in Table 3.1. It can be stated that:

- the exhaustive search returns 12 sets of sensors that allow the observability of the system;
- the obtained solutions are characterized by different values of the condition number (2.29);
- The third sensor set corresponds to the solution produced by the SA procedure, and possesses the lowest condition number value ($\mathcal{K}(W) = 4.1102 \times 10^3$).

#	Sensor sets	$\mathcal{K}(W)$
1	$[s_3,\ s_{13},\ s_{16},\ s_{17},\ s_{19},\ s_{20}]$	4.1208×10^3
2	$[s_3,\ s_{12},\ s_{16},\ s_{17},\ s_{19},\ s_{20}]$	4.1509×10^3
3	$[s_3,\ s_{13},\ s_{15},\ s_{17},\ s_{19},\ s_{20}]$	4.1102×10^3
4	$[s_3,\ s_{12},\ s_{15},\ s_{17},\ s_{19},\ s_{20}]$	4.1509×10^3
5	$[s_3,\ s_{13},\ s_{15},\ s_{16},\ s_{19},\ s_{20}]$	4.1719×10^3
6	$[s_3,\ s_{12},\ s_{15},\ s_{16},\ s_{19},\ s_{20}]$	4.1891×10^3
7	$[s_3,\ s_{13},\ s_{16},\ s_{17},\ s_{18},\ s_{20}]$	4.2209×10^3
8	$[s_3,\ s_{12},\ s_{16},\ s_{17},\ s_{18},\ s_{20}]$	4.3519×10^3
9	$[s_3,\ s_{13},\ s_{15},\ s_{17},\ s_{18},\ s_{20}]$	4.1369×10^3
10	$[s_3,\ s_{12},\ s_{15},\ s_{17},\ s_{18},\ s_{20}]$	4.1869×10^3
11	$[s_3,\ s_{13},\ s_{15},\ s_{16},\ s_{18},\ s_{20}]$	4.4511×10^3
12	$[s_3,\ s_{12},\ s_{15},\ s_{16},\ s_{18},\ s_{20}]$	4.1519×10^3

Table 3.1: Sensor sets computed via exhaustive search.

The results show that both the SA and GA techniques can identify a set of sensors that allow the system to be observable. They differ primarily in terms of computational speed and in regards of the cardinality of the sensor set. It can be seen that the SA approach finds smaller sensor sets than those found with GA, but requires slightly more computation time (GA). With respect to our case study, this discrepancy could be considered acceptable; however, in more extensive application scenarios where large number of sensors need to be considered, the choice of an efficient local search heuristic may be crucial.

An exhaustive search process was conducted with the goal of finding a possible alternative sensor set configuration with redundancy features. At the end of the process, the following sensor set configuration was selected

- Cardinality $\left|S^{alt}\right| = 6$;
- $\mathcal{S}^{alt} = [s_3,\ s_{12},\ s_{16},\ s_{17},\ s_{18},\ s_{20}]$.

It is observed that the set $\mathcal{S}^*_{sa}$ and $\mathcal{S}^{alt}$ differ as follows:

$$\left(\mathcal{S}^*_{sa} \cup \mathcal{S}^{alt}\right) \setminus \left(\mathcal{S}^*_{sa} \cap \mathcal{S}^{alt}\right) = \mathcal{S}^R = [s_{12}, s_{13}, s_{15}, s_{16}, s_{18}, s_{19}] \tag{3.48}$$

Then, the following augmented subset can be designed

$$\mathcal{S}^a = \mathcal{S}^*_{sa} \cup \mathcal{S}^R = [s_3,\ s_{12},\ s_{13},\ s_{15},\ s_{16},\ s_{17},\ s_{18},\ s_{19},\ s_{20}] \tag{3.49}$$

Note that the sensor triples $[s_{13}, s_{15}, s_{19}]$ and $[s_{12}, s_{16}, s_{18}]$ allows the definition of the redundant sensor configuration of interest.

3.4.3 Fault-tolerant Sensor Reconciliation Architecture Test

This test is used to demonstrate the effectiveness of the estimation architecture depicted in Figure 3.1. The proposed method is studied considering the asymptotically stable LTI model (3.1) with the system matrices (3.45) and (3.46) and the output matrix C defined according to the set of redundant sensors (3.49). To test the state reconstruction capabilities of the algorithm, it is also necessary that the sensors s_{12}, s_{15} and s_{18} are subjected to faults (loss of effectiveness $\gamma(k)$).

In addition, the system state that represents the number of vehicles per unit time in each sector is assumed to be bounded according to the traffic flow specified in UNI11248 (Appendix C) rules[2]. This consideration is exploited to define the Luenberger observer and reconciliator unit discussed in Section 3.2.1. In this respect, the system vector is considered to be bounded within the polytope

$$\mathcal{X} \coloneqq \{x : 0 \leq x \leq 15\}, \tag{3.50}$$

considering that the accounting street sector is classified as a local street. Moreover, it is also assumed that γ is confined within the polytope

$$\mathcal{Z} \coloneqq \gamma : \left[\underline{\gamma}^{(1)}, \underline{\gamma}^{(2)}, \underline{\gamma}^{(3)}\right]^T \leq \gamma \leq \left[\overline{\gamma}^{(1)}, \overline{\gamma}^{(2)}, \overline{\gamma}^{(3)}\right]^T \tag{3.51}$$

with $\underline{\gamma}^{(1)} = \underline{\gamma}^{(2)} = 0$, $\underline{\gamma}^{(3)} = 0.1$, $\overline{\gamma}^{(j)} = 1$, $j = 1, 2, 3$. As far as the numerical complexity of the method is of interest, observe that, for the presented case of study, it was necessary to calculate 2^{24} vertices, derived from the combination of those associated with $\hat{\gamma}$ (comprising 2^3 vertices) and those associated with $\hat{x}$ (comprising 2^{21} vertices). Solving the optimization problem (3.19)-(3.20), which determines the observer gains (3.13) for the given simulation scenario, it required an execution time of 805.81 [sec] with our hardware (workstation equipped with 2 $\times$ Intel® Xeon® Gold 5218 Processor 16-Core 2.3GHz and 12 $\times$ 32GB PC4-23400 2933MHz DDR4 RAM) However, it should be noted that the observer gains are calculated offline, while only the interpolation procedure (which requires extremely few computational resources) is performed in real-time. Assuming that the known signals shown in Figure 3.4 are used as input $u(k)$ and the unknown input $w(k)$ is a Gaussian white noise with zero mean and $Q_v = 0.1$, the ability of the proposed method to correctly reconstruct the system state even when sensor errors occur will be verified.

To make the estimation problem harder, it is also assumed also that a bias error $b(k)$ is present on the physical sensors s_{12}, s_{15} and s_{18} that consists of the following profile

$$b(k) = 1.5sin(0.3k). \tag{3.52}$$

In addition, gain faults affect the same sensors as shown in Figure 3.8. Figures 3.6-3.5 demonstrate the effectiveness of the proposed sensor reconciliation procedure. In particular, Figures 3.6 and 3.8 show the ability of the estimation architecture to estimate the bias error and the loss of effectiveness parameters. On the other hand, Figure 3.5 shows the performance of the whole architecture

[2]https://store.uni.com/uni-11248-2016

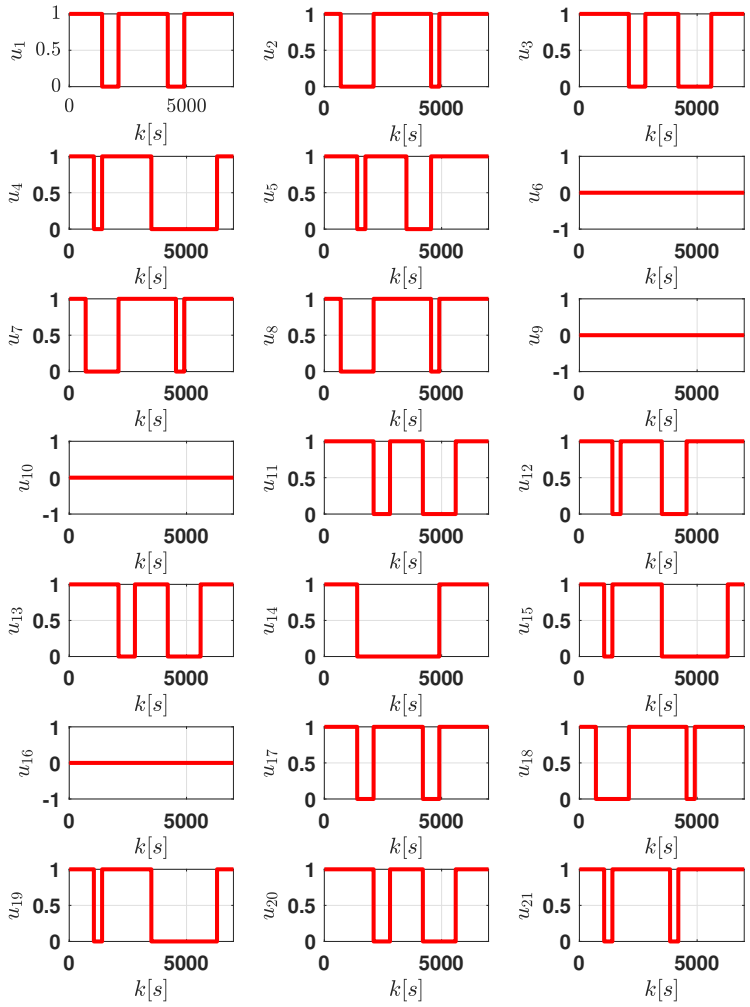

FIGURE 3.4: Simulative Test - Known signals used as traffic inputs $u_i(k)$, $i = 1, \ldots, 21$.

in terms of state reconstruction. In particular, the proposed sensor reconciliation scheme was compared with the state-of-the-art fault-hiding approach proposed in [94], where a reconfiguration unit is designed via a fuzzy Luenberger state observer. Figure 3.5 shows the comparison of the state reconstruction performance of the above mentioned estimation methods. In particular, the red dashed line (labeled as x) represents the evolution of the system state while its estimations are represented by the blue (proposed approach - $\hat{x}$) green (fault hiding approach proposed in [94] - $\hat{x}_{TS}$) and yellow (Luenberger approach - $\hat{x}_{std}$) lines.

The results show that the Luenberger observer perform poorly in reducing

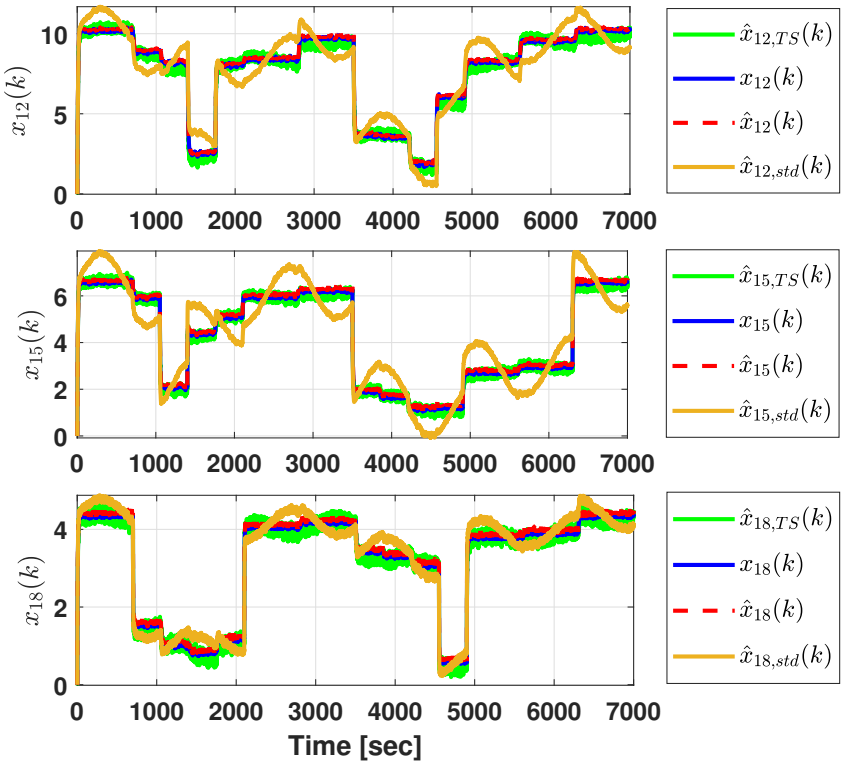

FIGURE 3.5: Simulative Test - State observer performance - System state (x_{12}, x_{15}, x_{18} - dashed red line) compared to its estimates ($\hat{x}_{12}$, $\hat{x}_{15}$, $\hat{x}_{18}$). Comparison of the proposed estimation architecture (blue line) with the Luenberger observer (yellow line) and the TS fuzzy approach (green line).

the effects of bias $b(k)$ and uncertain input $w(k)$, while the proposed estimation strategy and the TS approach provide better performance in terms of state reconstruction error. On the other side, Figure 3.9 reports an evaluation of the mentioned estimation methods in terms of the Mean Absolute Error (MAE). In this respect, the green (e_{TS}) blue (e) and yellow (e_{std}) bars refers to the fault-hiding [94], the proposed and the standard Luenberger approaches. The presented results show how the proposed sensor reconciliation scheme is able to outperform the other approaches in terms of state reconstruction MAE error.

3.4.4 Experimental Validation

To validate experimentally the presented approach, further studies have been undertaken by using the simulation framework software *Aimsun Next*[3], which is a traffic simulation software designed for modeling and analyzing transportation systems. Developed by TSS-Transport Simulation Systems, Aimsun Next allows microsimulations, network modeling, traffic demand modeling, dynamic

[3]https://www.aimsun.com/

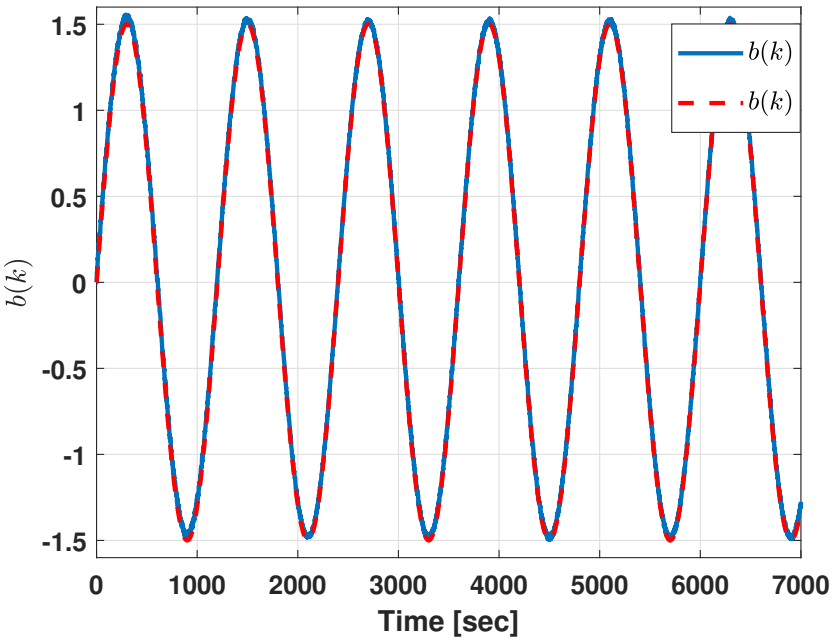

FIGURE 3.6: Simulative Test - Bias fault $b(k)$ (red dashed line) and its estimation $\hat{b}(k)$ (blue line).

FIGURE 3.7: Experimental validation using Aimusun Next software: roundabout (a) and intersection (b) designed for experimental validation.

traffic assignment, simulation output and analysis, calibration and validation, and so on. In particular, *Aimsun Next* is widely used by transportation engineers, planners, and researchers to simulate and evaluate the performance of traffic networks [95]–[99]. Regarding our case study, the *Aimsun Next* software has been used to create a realistic road network infrastructure (Figure 3.7(a)-(b)) and run simulations at a microscopic level by defining and considering the routes of each single cars. By exploiting all functionalities of the *Aimsun Next* framework it was possible to implement a sort of "digital twin" of the road and traffic scenario depicted in Figure 3.3. Two main steps were undertaken to generate the simulations. First, an identification procedure has been undertaken to iden-

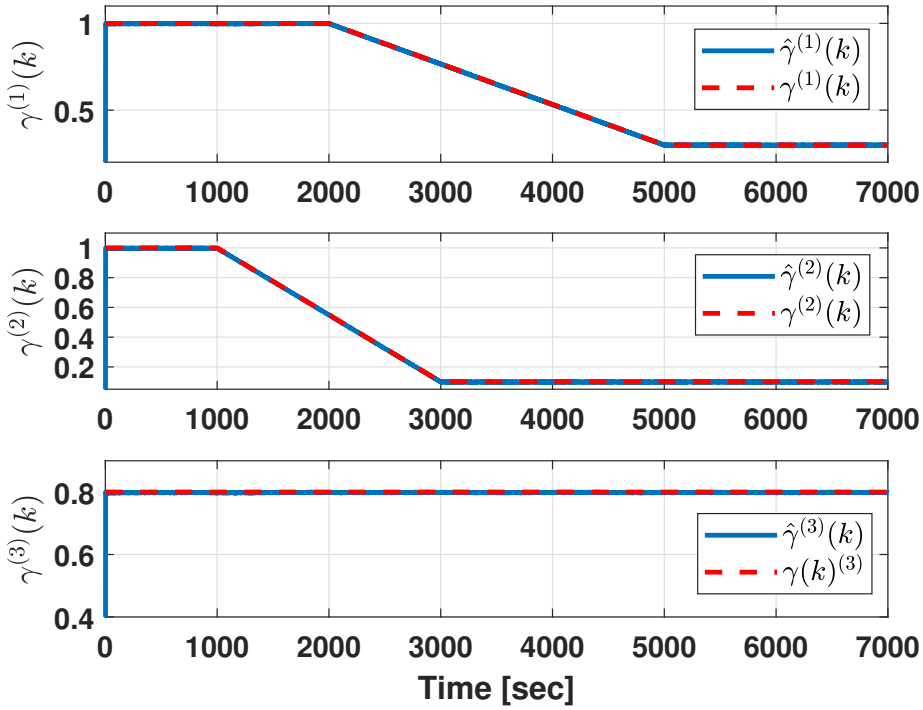

FIGURE 3.8: Simulative Test - Evolution of the loss of effectiveness parameters $\gamma^{(j)}$ (dashed red line) compared to its estimate $\hat{\gamma}^{(j)}$, $j = 1, 2, 3$ (blue line).

tify the traffic model corresponding to the traffic pattern generated by Aimsun Next. The identification activity has been performed by using the Prediction Error Minimization (PEM) algorithm provided by the Matlab System Identification Toolbox. More specifically, the PEM algorithm was used to update the initial model defined by the system matrices (3.45)-(3.46). In the second step, the SA procedure and the exhaustive search method have been used to find an augmented sensors subset $\mathcal{S}^a$. Interestingly enough, the same sensor subset $\mathcal{S}^a$ obtained in Section 3.4.2 has been found. The validation process included a test accounting for a collection of 12 hours of traffic data. Moreover, as in the simulation presented in Section 3.4.3, it was assumed that the physical sensors s_{12}, s_{15}, and s_{18} are subject to gain fault effects as shown in Figure 3.11. The validation test results are presented in Figures 3.10-3.11 and Table 3.2. In particular, Figure 3.10 compares the proposed estimation architecture with the *Aimsun Next* data, while Figure 3.11 shows the ability of the observer to estimate the loss of effectiveness parameters. Finally, Table 3.2 reports the Mean Absolute Errors computed over the 12-hour test period. The obtained results indicate that the proposed estimation architecture performs well also in this case.

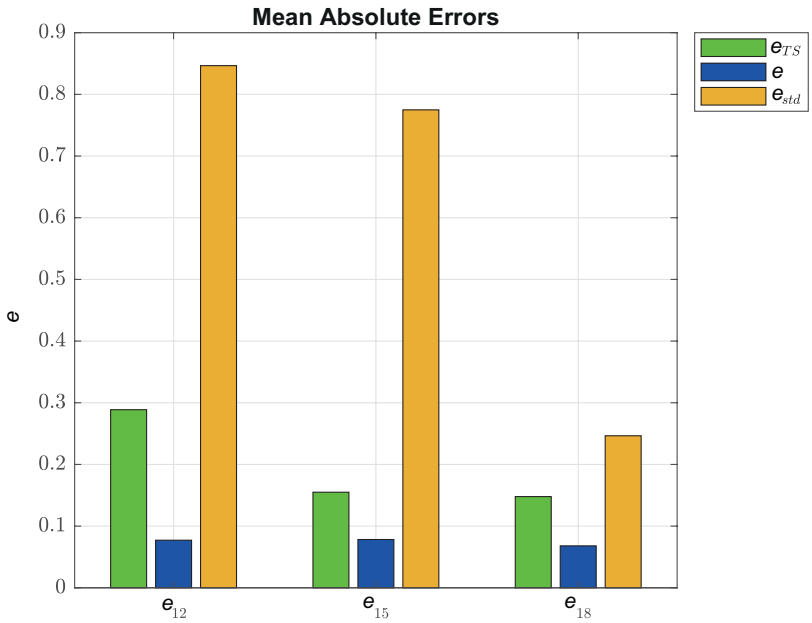

FIGURE 3.9: Simulative Test - Mean absolute errors computed for each estimation approach (e_{TS}, e, e_{std}).

$\mathbf{e_1}$	$\mathbf{e_2}$	$\mathbf{e_3}$	$\mathbf{e_4}$	$\mathbf{e_5}$	$\mathbf{e_6}$	$\mathbf{e_7}$
2.6	1.6	0.01	0.7	2.0	5.1	3.2
$\mathbf{e_8}$	$\mathbf{e_9}$	$\mathbf{e_{10}}$	$\mathbf{e_{11}}$	$\mathbf{e_{12}}$	$\mathbf{e_{13}}$	$\mathbf{e_{14}}$
1.9	2.3	6.6	0.1	0.9	0.5	0.6
$\mathbf{e_{15}}$	$\mathbf{e_{16}}$	$\mathbf{e_{17}}$	$\mathbf{e_{18}}$	$\mathbf{e_{19}}$	$\mathbf{e_{20}}$	$\mathbf{e_{21}}$
0.9	1.2	0.7	0.3	3.7	0.01	9.6

Table 3.2: Experimental validation - Mean absolute errors.

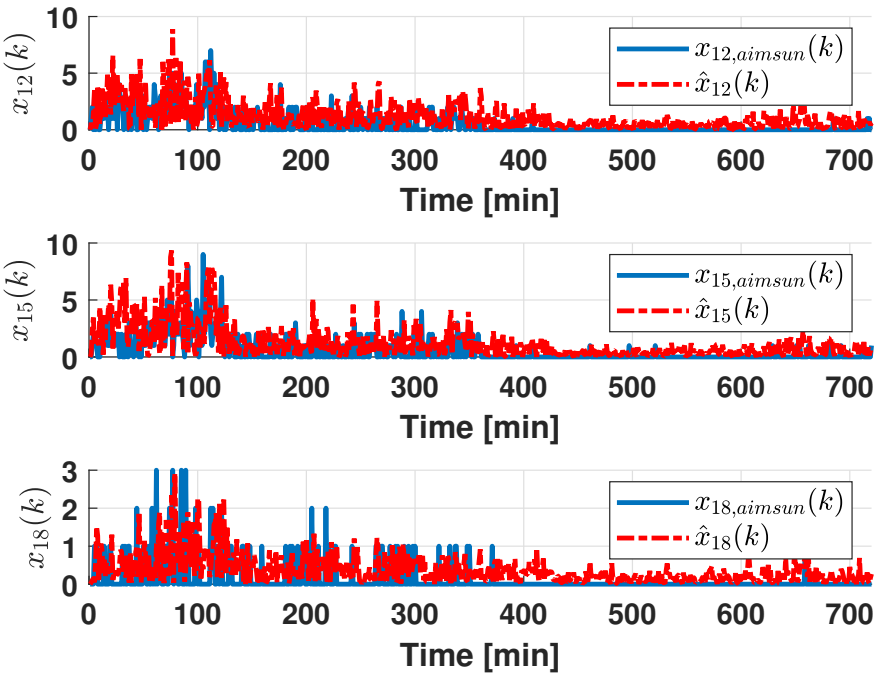

FIGURE 3.10: Experimental validation. State observer performance - Comparison of proposed estimation architecture ($\hat{x}_{12}$, $\hat{x}_{15}$, $\hat{x}_{18}$ - red dashed line) with the *Aimsun Next* data ($x_{12,aimsun}$, $x_{15,aimsun}$, $x_{18,aimsun}$ - blue line).

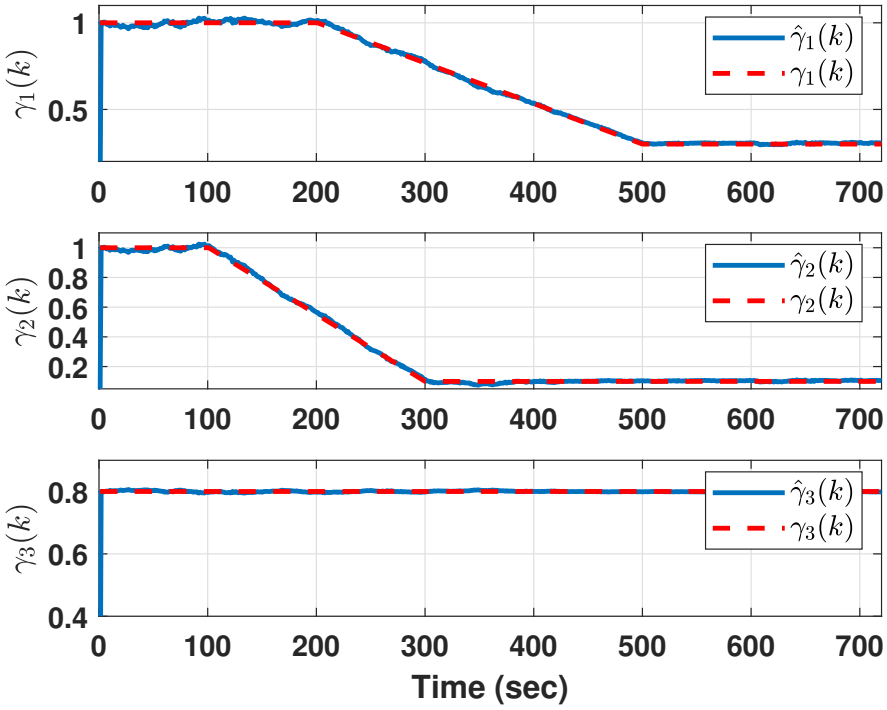

FIGURE 3.11: Experimental validation. Evolution of the loss of effectiveness parameters $\gamma^{(j)}$ (dashed red line) compared to its estimate $\hat{\gamma}^{(j)}$, $j = 1, 2, 3$ (blue line).

3.5 Conclusions

This Chapter has presented a fault-tolerant sensor reconciliation architecture that enables reliable WSNs. An LPV Luenberger observer and a reconciliation unit were used to develop the virtual sensor concept that form the foundation of the proposed solution. In particular, the overall architecture has allowed the reconstruction of the state of the system by hiding potentially corrupted measurements caused by sensor failures. The effectiveness of the proposed strategy was demonstrated by means of realistic simulation tests designed on the basis of a real infrastructure, using *Aimsun Next* as traffic load simulator. The proposed scheme also provides some methodological novelties to "fault-hiding" observer design problems. It is fair to point out that the proposed scheme has some limitations in its current level of development. A primary limitation is that the solution is centralized, in the sense that all data are collected to a central station where the computation are undertaken. For this reason the performance, in terms of computational complexity, of this scheme deteriorates in response to growing areas to be monitored and/or growing number of sensors. This issue is addressed in the next Chapter where a distributed architecture is presented.

Distributed Sensor Reconciliation Design 4

This Chapter investigates the sensor reconciliation problem from a distributed perspective. As a solution of the distributed state estimation problem, a Distributed Sensor Reconciliation (DSR) architecture is proposed that is grounded in a decomposed steady-state Kalman Filter (KF). The Chapter provides performance analysis of the architecture alongside proofs of its theoretical stability. To assess the performance of the DSR architecture, numerical simulations are presented and discussed. This Chapter is structured as follows: Section 4.1 introduces the DSR problem. Section 4.2 delves into the DSR design process, analyzing the steady-state centralized Kalman Filter and its decomposed formulation. The distributed filter is then used as foundation for the design of the distributed fault tolerant estimation architecture. As follow up, Section 4.3 discusses the architecture's stability. Numerical examples in Section 4.4 are used to evaluate the strategy's performance. Concluding remarks are offered in Section 4.5.

4.1 Problem Formulation

Lets consider the following discrete-time LTI system

$$x(k+1) = Ax(k) + Bu(k), \tag{4.1}$$

where: $k \in \mathbb{N}$, $x(k) \in \mathbb{R}^{n_x}$ is the system state and $u(k) \in \mathbb{R}^{n_u}$ a known exogenous input. A network consisting of m sensors is monitoring the above system. The measurement from each sensor $i \in \{1, \ldots, m\}$ is given by

$$y_i(k) = C_i(\gamma_i(k))x(k) + D_i u(k) + b_i(k) + v_i(k), \tag{4.2}$$

where $y_i(k) \in \mathbb{R}$ is a scalar, $D_i \in \mathbb{R}^{1\times n_u}$ is a known row vector and $v_i(k)$ is a Gaussian measurement noise. In a MAS context, each sensor can be considered as a node where all local computation take place for such a sensor. It is assumed that each sensor i is only able to communicate with a subset of other *neighboring* sensors and may be affected by sensor faults modeled by the n_x-dimensional row vector

$$C_i(\gamma_i(k)) = \gamma_i(k)C_i^h \in \mathbb{R}^{1\times n_x} \tag{4.3}$$

where C_i^h is the healthy output matrix of sensor i and $\gamma_i \in [0,1]$ is its effectiveness (multiplicative fault). On the other hand, the scalar $b_i(k)$ models additive faults (bias). It is assumed that sensors cannot regain their effectiveness over time; therefore, $\gamma(k)$ is monotonically non-increasing. It is clear that if sensors are not affected by faults one would have $\gamma_i(k) = 1$ and $b_i(k) = 0$ for $i = 1, \ldots, m$ (*global healthy condition*). By stacking the measurement equations the following aggregated output is calculated

$$y(k) = C(\gamma(k))x(k) + Du(k) + b(k) + v(k), \tag{4.4}$$

where

$$\begin{aligned} y(k) &\coloneqq \begin{bmatrix} y_1(k) \\ \vdots \\ y_m(k) \end{bmatrix}, D \coloneqq \begin{bmatrix} D_1 \\ \vdots \\ D_m \end{bmatrix}, b(k) \coloneqq \begin{bmatrix} b_1(k) \\ \vdots \\ b_m(k) \end{bmatrix}, \\ \gamma(k) &\coloneqq \begin{bmatrix} \gamma_1(k) \\ \vdots \\ \gamma_m(k) \end{bmatrix}, C(\gamma(k)) \coloneqq \begin{bmatrix} C_1(\gamma_1(k)) \\ \vdots \\ C_m(\gamma_m(k)) \end{bmatrix}, \end{aligned} \tag{4.5}$$

and $v(k) = [v_1(k) \ldots, v_m(k)]^T$ is a zero-mean independent and identically distributed (i.i.d) Gaussian noise with covariance $Q_v \geq 0$. This enables one to assume that despite potential fault occurrences, the observability of the system is preserved i.e. $(A, C(\gamma(k)))$ is observable $\forall k > 0$.

Please note that the local observability of $(A, C_i(\gamma_i(k)))$ is not guaranteed. In other words, a single sensor may not be capable of observing the entire state space, or it may not be functioning at all ($\gamma_i(k) = 0$). To avoid any loss of observability, redundant sensors are typically considered to ensure that each component of the state is measured by multiple sensors. It is also assumed perfect estimation of the multiplicative fault, enabling one to utilize the fault parameter $\gamma(k)$ as a known quantity. While these assumption may not reflect real-world scenarios, similar or more restrictive assumptions are not uncommon in the literature [68]. Nevertheless, to address this limitation, existing estimation schemes [100], [101] are currently being investigated and will be the focus of future research. The main Distributed Sensor Reconciliation Design problem can be stated as follows:

Problem 4.1.1. ***(DSR Design Problem):*** *Given system (4.1), despite the presence of unknown additive faults $b(k)$, known multiplicative faults $\gamma(k)$ and the disturbance $v(k)$, calculates the state estimate $\hat{x}(k)$ locally at each time $k \geq 0$ and for each sensor $i = 1, \ldots, m$ relaying only on the local measurement $y_i(k)$ and neighboring data.*

4.2 Design of the Distributed Sensor Reconciliation Architecture

This work is inspired by the distributed decomposition of a steady-state Kalman filter proposed in [63] for LTI autonomous systems. However, direct application of this decomposition to our problem is not feasible for two main reasons: (i)

system (4.1) is non-autonomous; and (ii) it does not account for additive and multiplicative faults. To address the first issue, considering that the input $u(k)$ is known, the idea it to decompose the state $x(k)$ into its natural component $x^n(k)$ and its forced component $x^f(k)$ [102] as

$$x(k) = x^n(k) + x^f(k), \tag{4.6}$$

and employ the distributed Kalman filter to estimate solely the natural component of the state. Consequently, the output can also be decomposed into a natural and forced component, calculated from the real output as

$$y^f(k) = C(\gamma(k))x^f(k) + Du(k) \tag{4.7}$$
$$y^n(k) = y(k) - y^f(k). \tag{4.8}$$

The overall state is reconstructed by adding the natural component to the forced component, which is calculated using information derived from the known input as

$$x^f(k) = \sum_{\tau=0}^{k-1} A^{k-\tau-1}Bu(\tau), \quad k > 0. \tag{4.9}$$

Finally, the system used to design the distributed Kalman filter is

$$x^n(k+1) = Ax^n(k) \tag{4.10}$$
$$y^n(k) = C(\gamma(k))x^n(k) + b(k) + v(k). \tag{4.11}$$

4.2.1 Centralized Kalman Filter

Before presenting the decomposed filter, let's analyze the impact of multiplicative faults $\gamma(k)$ on the filter design from a centralized perspective. Given the extended operation duration of typical sensor networks, it's reasonable to assume that the Kalman filter operates in a steady state condition, resulting in a steady-state Kalman filter with a fixed gain [103]

$$K_{\gamma(k)} = P_{\gamma(k)}C_{\gamma(k)}^T(C_{\gamma(k)}^T P_{\gamma(k)}C_{\gamma(k)} + Q_v)^{-1}, \tag{4.12}$$

$P_{\gamma(k)}$ represents the steady-state error covariance of the estimate provided by the Kalman filter. To simplify notation the terms $C_{\gamma(k)} = C(\gamma(k))$ and $K_{\gamma(k)} = K(\gamma(k))$ have been used.

Remark 4.2.1. *It's important to note that (4.12) holds if $\gamma(k)$ remains constant until the Kalman filter reaches its steady-state condition.*

Accordingly, the optimal Kalman estimate is computed recursively as

$$\hat{x}^n(k+1) = A\hat{x}^n(k) + K_{\gamma(k)}(y^n(k+1) - C_{\gamma(k)}A\hat{x}^n(k)) \tag{4.13}$$
$$= [A - K_{\gamma(k)}C_{\gamma(k)}A]\hat{x}^n(k) + K_{\gamma(k)}y^n(k+1). \tag{4.14}$$

It is clear that the system matrices of the Kalman estimate are closely connected to the effectiveness $\gamma(k)$, resulting in a linear time-varying dynamical system.

To estimate the state $x^n(k)$ while taking into account $\gamma(k)$ the main idea is to develop a switching Kalman filter whose switches are determined by the value of the multiplicative fault $\gamma(k)$. Define the quantized effectiveness $q(k)$ as

$$q(k) \coloneqq f_q(\gamma(k)), \tag{4.15}$$

where $f_q(\cdot)$ is a quantizer which is static and time-invariant. The set of (distinct) quantized levels is

$$\mathcal{O} \coloneqq \{0, \rho_q, 2\rho_q, \ldots, (n_o - 1)\rho_q\} \cup \{1\}, \tag{4.16}$$

where ρ_q is the quantization step and $n_o = card(\mathcal{O})$ is the number of quantized levels. Each quantization level is associated to a segment, such that the quantizer maps the entire segment to that specific quantization level. Moreover, these segments collectively form a partition of the interval $[0, 1]$, indicating they are disjoint and their union equals $[0, 1]$. The set of all possible faulty conditions, i.e. all combinations with repetition of the quantized levels in $\mathcal{O}$ in m values, is

$$\mathcal{Q} \coloneqq \left\{\bar{q}_1, \ldots, \bar{q}_{n_q}\right\}, \tag{4.17}$$

where $n_q = card(\mathcal{Q})$ and $\bar{q}_i \in \mathbb{R}^m$ for $i = 1, \ldots, n_q$, characterize the possible faulty configurations. A family of parameterized Kalman gains and the associated faulty output matrix are then calculated as

$$\bar{\mathcal{K}} \coloneqq \{\bar{K}_1, \ldots, \bar{K}_{n_q}\} \tag{4.18}$$

$$\bar{\mathcal{C}} \coloneqq \{\bar{C}_1, \ldots, \bar{C}_{n_q}\}, \tag{4.19}$$

where $\bar{K}_i = K(\bar{q}_i)$ and $\bar{C}_i = C(\bar{q}_i)$ are the Kalman gain and the faulty output matrix related to the faulty condition $\bar{q}_i$ for $i = 1, \ldots, n_q$. As a result, the following gain scheduling Kalman filter is formulated

$$\begin{aligned} \hat{x}^n(k+1) = [A - \bar{K}_{\sigma(k)}\bar{C}_{\sigma(k)}A]\hat{x}^n(k) \\ + \bar{K}_{\sigma(k)}y^n(k+1) + \epsilon_q(k), \end{aligned} \tag{4.20}$$

where $\sigma : \mathbb{N} \to \mathcal{I}$, with $\mathcal{I} \coloneqq \{1, \ldots, n_q\}$, is the switching signal that selects, at each time k, the Kalman gain and the output matrix associated to the current value of the quatized effectiveness $q(k)$ and $\epsilon_q(k)$ is the quantization error.

4.2.2 Filter Decomposition

In this Section, the main steps and concepts related to the decomposition of the Kalman filter are briefly discussed. Given that the closed-loop matrices of the filters $A - \bar{K}_l\bar{C}_lA$ are asymptotically stable $\forall l \in \mathcal{I}$, it is always possible to find a set of matrices $\{\Lambda_l \,|\, \forall l \in \mathcal{I}\}$, where $\Lambda_l \in \mathbb{R}^{n_x \times n_x}$ is a strictly stable, non-derogatory Jordan matrix with the same characteristic polynomial of $A - \bar{K}_l\bar{C}_lA$. Furthermore, it has been proved in [63] that the pair $(\Lambda_l, \mathbf{1}_{n_x})$ is controllable [1].

[1] $\mathbf{1}_{n_x}$ denotes the column vector in $\mathbb{R}^{n_x}$ whose elements are all ones.

As a consequence, it is always possible to calculate matrices $F_l^i \in \mathbb{R}^{n_x \times n_x}$ with $i = 1, \ldots, m$, such that the following equalities hold:

$$\begin{aligned} F_l^i \Lambda_l &= (A - \bar{K}_l \bar{C}_l A) F_l^i \\ F_l^i \mathbf{1}_{n_x} &= \bar{K}_l^i, \end{aligned} \tag{4.21}$$

where $\bar{K}_l = \left[\bar{K}_l^1, \ldots, \bar{K}_l^m\right]$. To guarantee the stability of the local filters when the matrix A has unstable eigenvalues the following matrices are introduced

$$S_l = \Lambda_l + \mathbf{1}_{n_x} \beta_l^T, \quad \forall l \in \mathcal{I}, \tag{4.22}$$

where the vectors $\beta_l \in \mathbb{R}^{n_x}$ are computed so that the matrices S_l have the same unstable eigenvalues of A, while the stable eigenvalues are freely chosen but cannot coincide with those of Λ_l. Note that if A is strictly stable, β_l can be chosen as a zero vector. The following family of triplets is computed

$$\mathcal{F} \coloneqq \{(F_l, S_l, \beta_l) \mid \forall l \in \mathcal{I}\}, \tag{4.23}$$

where $F_l = \left[F_l^1, \ldots, F_l^m\right] \in \mathbb{R}^{n_x \times m n_x}$. The set $\mathcal{F}$ stores, for each value $l \in \mathcal{I}$ and hence for each faulty configuration, the associated matrices F_l, S_l and the vectors β_l computed as in (4.21) and (4.22).

Finally, the centralized Kalman filter can be decomposed into m local filters (one for each output $y_i(k)$) whose dynamics are given by

$$\begin{aligned} z_i(k) &= y_i^n(k+1) - \beta_{\sigma(k)} \hat{\xi}_i(k) \\ \hat{\xi}_i(k+1) &= S_{\sigma(k)} \hat{\xi}_i(k) + \mathbf{1}_{n_x} z_i(k), \end{aligned} \tag{4.24}$$

where $z_i(k) \in \mathbb{R}$ is a local residual and $\hat{\xi}_i(k)$ is the estimated decomposed local state. Just like in (4.20), the same switching signal $\sigma(k)$ is utilized to pick the active triplet from $\mathcal{F}$ to reconfigure and mitigate the impact of a "meaningful" change on $\gamma(k)$. In conclusion, the estimated state can be calculated by summing the decomposed state as follows

$$\hat{x}^n(k+1) = \sum_{i=1}^{m} F_{\sigma(k)}^i \hat{\xi}_i(k+1) = F_{\sigma(k)} \hat{\xi}_i(k+1). \tag{4.25}$$

The proof that (4.24) and (4.14) share the same input-output relationship, $z_i(k)$ is bounded and (4.25) holds can be found in [63].

4.2.3 Synchronization Algorithm

The output of each sensor is collected by a node that computes the estimate $\hat{\xi}_i(k)$ using a local filter, whose dynamics are outlined in (4.24). The interactions among nodes are modeled by means of an undirected weighted graph $\mathcal{G} \coloneqq (\mathcal{V}, \mathcal{E}, \mathcal{A})$ where $\mathcal{V} \coloneqq \{1, \ldots, m\}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the set of edges, and $\mathcal{A} \coloneqq [a_{ij}] \in \mathbb{R}^{m \times m}$ is the adjacency matrix whose elements satisfy $a_{ii} = 0$ and $a_{ij} > 0$ if and only if $(i, j) \in \mathcal{E}$. The matrix $\mathcal{L} \in \mathbb{R}^{m \times m}$ is the Laplacian matrix of $\mathcal{G}$, which is positive semi-definite and with exactly one zero eigenvalue. For each node i, the set of its neighbors is denoted as $\mathcal{V}_i \coloneqq \{j \mid (i, j) \in \mathcal{E}\}$ and its cardinality is $n_i \coloneqq card(\mathcal{V}_i)$.

Remark 4.2.2. *From a practical point of view, sensors spatially close to each other can be associated to a single augmented node composed by duplicated virtual nodes running on the same hardware and thus sharing the same communication links.*

To compute the local state estimate $\check{x}_i(k)$, each node i uses neighborhood data to acquire $\hat{\xi}_j(k)$ for all $j \in \mathcal{V}$. This is accomplished by introducing the aggregated state

$$\eta_i(k) \coloneqq \left[\eta_{ij}^T(k), \ldots, \eta_{im}^T(k)\right]^T \in \mathbb{R}^{mn_x}, \tag{4.26}$$

where $\eta_{ij}(k)$ is the inference that node i makes on $\hat{\xi}_j(k)$ and updating it by means of the following synchronization algorithm

$$\eta_i(k) = \tilde{S}_{\sigma(k)}\eta_i(k) + \tilde{L}_i z_i(k) + \tilde{B} g_i(k), \quad \forall i \in \mathcal{V}, \tag{4.27}$$

where[2]: $\tilde{S}_{\sigma(k)} = I_m \otimes S_{\sigma(k)}$, $\tilde{L}_i = e_i^m \otimes \mathbf{1}_{n_x}$ and $\tilde{B} = I_m \otimes \mathbf{1}_{n_x}$. Note that, at each instant k, the synchronization algorithm uses the same matrix $S_{\sigma(k)}$ selected in (4.24), thus preserving coherence between the inference $\eta_{ii}(k)$ and the decomposed state $\hat{\xi}_i(k)$. The (consensus) input $g_i(k)$ is designed as

$$g_i(k) = \sum_{j=1}^{m} a_{ij}(\Delta_j(k) - \Delta_i(k)), \tag{4.28}$$

where the vector $\Delta_i(k) \coloneqq \tilde{\Gamma}_{\sigma(k)}\eta_i(k) \in \mathbb{R}^m$ for $i \in \mathcal{V}$ is the message that sensor i sends at time k to its neighbors. The matrices $\tilde{\Gamma}_l = I_m \otimes \Gamma_l$ for $l \in \mathcal{I}$ are selected from an additional set $\mathcal{T} \coloneqq \{\Gamma_l | \forall l \in \mathcal{I}\}$ and each element of the set is computed by solving the modified Riccati inequality (4.30).

Lemma 4.2.1. *For a constant switching signal $\sigma(k) = l$, systems (4.27) reach consensus under the action of the consensus input (4.28) with Γ_l computed as follows*

$$\Gamma_l = \frac{2}{\lambda_2 + \lambda_m} \frac{\mathbf{1}_{n_x}^T P_l S_l}{\mathbf{1}_{n_x}^T P_l \mathbf{1}_{n_x}} \in \mathbb{R}^{1 \times n_x}, \tag{4.29}$$

where λ_2 and λ_m are, respectively, the second smallest and largest eigenvalues of $\mathcal{L}$. Moreover, $P_l = P_l^T > 0$ is the solution of the following modified algebraic Riccati inequality

$$P_l - S_l^T P_l S_l + (1 - \zeta^2)\frac{S_l^T P_l \mathbf{1}_{n_x} \mathbf{1}_{n_x}^T P_l S_l}{\mathbf{1}_{n_x}^T P_l \mathbf{1}_{n_x}} > 0, \tag{4.30}$$

with ζ satisfying $\prod_j |\lambda_j^u(S_l)| < \zeta^{-1} \leq \frac{1+\lambda_2/\lambda_m}{1-\lambda_2/\lambda_m}$, where $\lambda_j^u(S_l)$ are the unstable eigenvalues of S_l.

[2] e_i^m denotes the i-th canonical basis vector in $\mathbb{R}^m$ and $\otimes$ is the Kronecher product.

Proof. This proof follows from the proof given in [63] and the fundamental steps are here reported for completeness. Analyze a generic vertex l. For any $j \in \{2, \dots, m\}$, let one denote $\zeta_j = \frac{1-2\lambda_j}{\lambda_2+\lambda_m} \leq \zeta$. Since $(S_l, \mathbf{1}_{n_x})$ is controllable $\forall l$, there exist some matrices $P_l > 0$, which solve (4.30). Together with , (4.29) it holds that

$$(S_l - \lambda_j \mathbf{1}_{n_x}\Gamma_l)^T P_l (S_l - \lambda_j \mathbf{1}_{n_x}\Gamma_l) \tag{4.31}$$

$$= S_l^T P_l S_l - (1-\zeta^2)\frac{S_l^T P_l \mathbf{1}_{n_x}\mathbf{1}_{n_x}^T P_l S_l}{\mathbf{1}_{n_x}^T P_l \mathbf{1}_{n_x}} - P_l < 0 \tag{4.32}$$

$$\leq S_l^T P_l S_l - (1-\zeta^2)\frac{S_l^T P_l \mathbf{1}_{n_x}\mathbf{1}_{n_x}^T P_l S_l}{\mathbf{1}_{n_x}^T P_l \mathbf{1}_{n_x}} - P_l < 0, \quad \forall l \in \mathcal{I}. \tag{4.33}$$

The above inequalities hold for each vertex l, thus concluding the proof. ■

In virtue of Lemma 4.2.1, under the action of (4.28), for constant switching signals the switching systems (4.27) exponentially achieve consensus in the mean-square sense, i.e., there exist $c > 0$ and $p \in (0, 1)$ such that

$$\mathbb{E}\left[\|\eta_i(k) - \bar{\eta}(k)\|^2\right] < cp^k \quad \forall i \in \mathcal{V}. \tag{4.34}$$

where $\mathbb{E}[\cdot]$ denotes the expected value and $\bar{\eta}(k)$ is the average value at time k of the aggregated states $\eta_i(k)$ for $i = 1, \dots, m$.

In conclusion, each agent i calculates a local estimate $\check{x}_i^n(k)$ by

$$\frac{1}{m}\sum_{i=1}^{m} \check{x}_i^n(k) = mF_{\sigma(k)}\eta_i(k) = \hat{x}^n(k), \quad \forall k > 0, \tag{4.35}$$

It is finally worth pointing out that the average of all local estimates coincides with the optimal Kalman estimate [63].

4.2.4 Estimation of Additive Faults

Additive faults are closely linked to the signal $b(k)$. Specifically, when $b_i(k) > 0$, the output of sensor i is affected by an additive fault. To address this type of fault, the idea is to utilize the sensor output $y_i(k)$ to compute an estimate of $b_i(k)$ and employ such an estimated value, say $\hat{b}_i(k)$, to mitigate the faulty action. The estimation of the bias signal is performed by a batch least-mean-squares approach [104] within a windowed data processing strategy. This method detects constant or slowly-varying additive faults. By estimating the local value $\hat{b}_i(k)$ for each sensor, our aim is to determine a scalar $\hat{b}_i(k)$ such that the estimated output $\hat{y}(k)$ closely matches the measured signals of the plant over the last N time steps. It is assumed that the last N samples of both the physical outputs $y(k)$ and the state estimation $\hat{x}(k)$ of the system are available at the current time step k. Under the *equivalence hypothesis* $\hat{x}(k) = x(k)$, the following consistency equations can be established:

$$y_i^n(k+1-j) = C_i(\gamma_i(k))A\hat{x}^n(k-j) + \hat{b}_i(k+1) \; j = 0, \dots, N-1. \tag{4.36}$$

As a consequence, $\hat{b}(k+1)$ is computed by solving the unconstrained optimization problem

$$b^\star = \arg\min_p \frac{1}{2} \sum_{j=0}^{N-1} |y_i^n(k+1-j) - C_i(\gamma(k+1-j))A\hat{x}^n(k-j) - p|_2^2, \quad (4.37)$$

where $N > 0$ is the estimation interval and $\hat{b}_i(k+1) = b^\star$. As the rate of change of $b(k)$ over time decreases, the estimation of the bias $b(k)$ at time k becomes more accurate for higher values of N.

4.2.5 Distributed Sensor Reconciliation Architecture

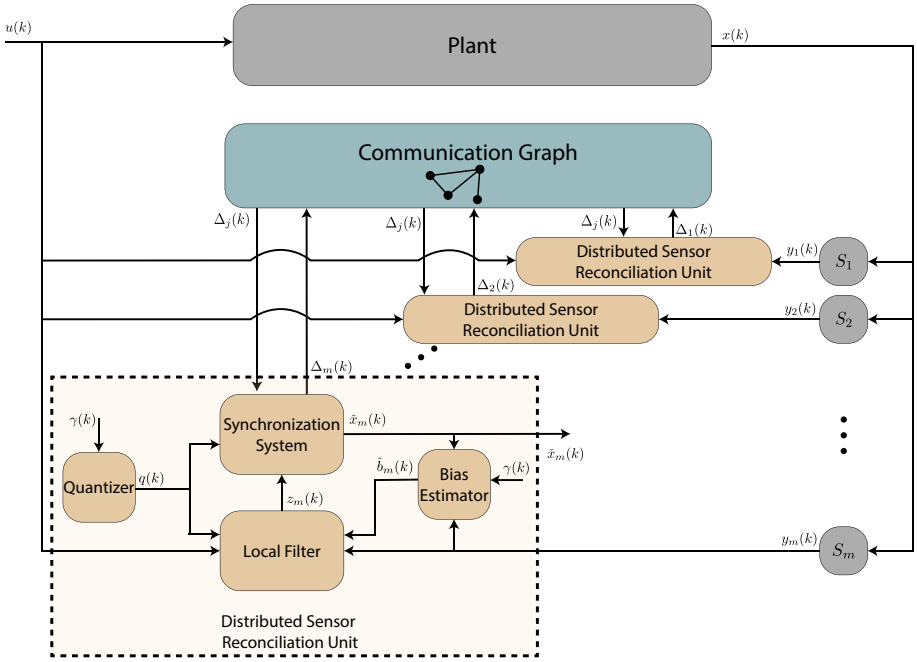

FIGURE 4.1: Distributed Sensor Reconciliation architecture.

Figure 4.1 provides a visual representation of the DSR architecture. Each sensor i is linked to a DSR unit consisting of four essential components:

- A *local filter* that employs eq. (4.24) to estimate the locally observable components of the state;
- A *synchronization module* that combines local information with data coming from neighboring units (represented in Figure 4.1 with the generic quantity $\Delta_j(k)$) to reconstruct the local state estimation $\check{x}_i(k)$ by means of eq. (4.27);
- An *estimator* responsible for solving the optimization problem (4.37) and thus determine an estimation of the bias value $b_i(k)$;

- A *quantizer* that calculates the quatized effectiveness $q(k)$ based on the current value of $\gamma(k)$.

The algorithm of the DSR architecture comprises an offline phase and an online phase. During the offline phase, the sets of matrices $\mathcal{F}$ and $\mathcal{T}$ are computed. The matrices S_l, F_l, Γ_l, and the vectors β_l are determined by following the procedure outlined in Sections 4.2.2 and 4.2.3.

During the online phase, the forced component of the state $x^f(k+1)$ is calculated as shown in (4.9) and used to calculate the natural component of the output. Subsequently, each sensor i utilizes the estimated value $\hat{b}_i(k+1)$ to compensate for the actual bias affecting the output.

It is worth point out that the most computational demanding tasks (computation of the set $\mathcal{F}$ and $\mathcal{T}$) are completed during the offline phase. The detailed procedure is presented in Algorithm 1.

Algorithm 1: Distributed Sensor Reconciliation

1: Estimate $\hat{b}(k+1)$ by solving (4.37) and compute $q(k) = f(\gamma(k))$ and update the local estimate by

$$\begin{aligned} z_i(k) &= (y_i^n(k+1) - \hat{b}_i(k+1)) - \beta_{\sigma(k)}\hat{\xi}_i(k) \\ \hat{\xi}_i(k+1) &= S_{\sigma(k)}\hat{\xi}_i(k) + \mathbf{1}_{n_x} z_i(k), \end{aligned} \tag{4.38}$$

2: Compute $\Delta_i(k) = \tilde{\Gamma}_{\sigma(k)}\eta_i(k)$, collect $\Delta_j(k)$ from neighbors and execute the consensus algorithm as

$$\eta_i(k+1) = \tilde{S}_{\sigma(k)}\eta_i(k) + \tilde{L}_i z_i(k) + \tilde{B}\sum_{j=1}^{m} a_{ij}(\Delta_j(k) - \Delta_i(k)), \tag{4.39}$$

where $\tilde{S}_{\sigma(k)} = I_m \otimes \tilde{S}_{\sigma(k)}$ and $\tilde{\Gamma}_{\sigma(k)} = I_m \otimes \Gamma_{\sigma(k)}$.

3: Update the fused estimate as

$$\check{x}_i(k+1) = mF_{\sigma(k)}\eta_i(k+1) + x^f(k+1).$$

4: Transmit the new state $\Delta_i(k+1)$ to neighbors.

where $\eta_i(0) = \mathbf{0}_{mn_x}$, $\xi_i(0) = \mathbf{0}_{n_x}$ for all $i = 1, \ldots, m$.

Remark 4.2.3. *Adding redundant sensors increases the computational overhead of the DSR approach. Specifically, virtual nodes representing local redundancy increase only local computational complexity, whereas redundant physical nodes introduce additional communication overhead.*

4.3 Stability Analysis

The dynamics that characterize a DSR unit (eqs. (4.38) and (4.39)) exhibit switching behaviors. In fact, whenever a new multiplicative fault configuration is detected, a new operative mode is selected in order to adapt the set-up to

the new faulty condition. In this Section, leveraging the assumption that sensor cannot recover effectiveness over time, the stability of the overall architecture is proven. Without loss of generality, let the system matrix be partitioned as

$$A = \begin{bmatrix} A^u & \\ & A^s \end{bmatrix}, \tag{4.40}$$

where $A^u \in \mathbb{R}^{n_x^u \times n_x^u}$ and $A^s \in \mathbb{R}^{n_x^s \times n_x^s}$, such that the eigenvalue of A^u lies on or outside the unit circle while the eigenvalues of A^s are strictly within the unit circle. The state and the output matrix $C(\gamma(k))$ can be accordingly rearranged as $x(k) = [x^s(k), x^u(k)]$ and $C(\gamma(k)) = [C^s(\gamma(k)), C^u(\gamma(k))]$. As a consequence, also the output matrices in $\bar{\mathcal{C}}$ can be rearranged in a similar way. Define the following matrices for $l = 1, \ldots, n_q$

$$L_\delta \coloneqq \left[I_m - \frac{1}{m} \mathbf{1}_m \mathbf{1}_m^T \right] \otimes I_{mn_x} L_\eta \tag{4.41}$$

$$A_l^{\tilde{\delta}} \coloneqq blkdiag(\tilde{S}_l, \tilde{S}_l - \lambda_2 \tilde{B}\tilde{\Gamma}_l, \ldots, \tilde{S}_l - \lambda_m \tilde{B}\tilde{\Gamma}_l) \tag{4.42}$$

$$A_l^{\epsilon} \coloneqq -(I_m \otimes \mathbf{1}_{n_x}) \bar{C}_l^s A^s, \tag{4.43}$$

with $L_\eta \coloneqq blkdiag(\tilde{L}_1, \ldots, \tilde{L}_m)$. With the above notions, it is possible to give the following proposition.

Proposition 4.3.1. *Suppose that* $\prod_j |\lambda^u(S_l)| < \frac{1+\lambda_2/\lambda_m}{1-\lambda_2/\lambda_m}$ *for all* $l \in \mathcal{I}$. *Then, Algorithm 1 guarantees that the estimation error* $\check{e}_i(k)$ *for each sensor* i *is bounded at any time* k.

Proof. Lets consider the estimation error for each sensor i

$$\begin{aligned} \check{e}_i(k) &\coloneqq \check{x}_i(k) - x(k) \\ &= (\check{x}_i(k) - \hat{x}(k)) + (\hat{x}(k) - x(k)) \\ &= \bar{e}_i(k) + \hat{e}(k), \end{aligned} \tag{4.44}$$

where $\hat{e}(k)$ is the estimation error of the Kalman filter and $\bar{e}_i(k)$ is an additional error caused by the synchronization algorithm. Lets study the boundedness of the two errors individually. The synchronization error can be expressed as

$$\bar{e}_i(k) = mF_{\sigma(k)}(\eta_i(k) - \bar{\eta}(k)) = mF_{\sigma(k)}\delta_i(k), \tag{4.45}$$

where $\delta_i(k)$ represents the consensus state deviation for sensor i. Stacking the errors $\bar{e}_i(k)$ from all sensors, it is found that

$$\bar{e}(k) = (I_m \otimes mF_{\sigma(k)})\delta(k), \tag{4.46}$$

where $\delta(k) = col(\delta_1, \ldots, \delta_m)$. The evolution of the aggregated state deviation can be expressed as

$$\delta(k+1) = \left[I_m \otimes S_{\sigma(k)} - \mathcal{L}(\tilde{B}\tilde{\Gamma}_{\sigma(k)}) \right] \delta(k) + L_\delta z(k), \tag{4.47}$$

Recall that the Laplacian matrix of an undirected graph is symmetric. Therefore, a unitary matrix Θ always exist such that

$$blkdiag(0, \lambda_2, \ldots, \lambda_m) = \Theta^T \mathcal{L} \Theta. \tag{4.48}$$

Finally, the following quantity can be introduced[3]

$$\tilde{\delta}(k) = (I_m \otimes \Theta)\delta(k). \tag{4.49}$$

Then, the boundedness of $\bar{e}_i(k)$ is related to the stability of system

$$\begin{bmatrix} \tilde{\delta}(k+1) \\ \epsilon(k+1) \\ x^s(k+1) \end{bmatrix} = H_{\sigma(k)} \begin{bmatrix} \tilde{\delta}(k) \\ \epsilon(k) \\ x^s(k) \end{bmatrix} + d_1(k), \tag{4.50}$$

where $d_1(k)$ collects all those quantities related to the measurement noise $v(k)$, $\epsilon(k)$ is an additional state component that models the evolution of the bounded residual $z(k)$ and $H_{\sigma(k)}$ is given below

$$H_{\sigma(k)} = \begin{bmatrix} A^{\tilde{\delta}}_{\sigma(k)} & L_{\tilde{\delta}}(I_m \otimes \beta_{\sigma(k)}) & L_{\tilde{\delta}} \bar{C}^s_{\sigma(k)} A^s \\ & I_m \otimes \Lambda_{\sigma(k)} & A^{\epsilon}_{\sigma(k)} \\ & & A^s \end{bmatrix}. \tag{4.51}$$

A detailed procedure on how to derive (4.50) is reported in [63] (see Appendix F). Furthermore, in [63] it is proven that if $\prod_j |\lambda^u(S_l)| < \frac{1+\lambda_2/\lambda_m}{1-\lambda_2/\lambda_m}$ for all $l \in \mathcal{I}$, solutions of (4.30) exist $\forall l \in \mathcal{I}$ and every mode of the switching system (4.50) is stable. Recalling that the effectiveness $\gamma_i(k)$ of sensor i cannot increase over time, it's straightforward to demonstrate that the number of switches is finite. In fact, after each switch, the number of possible faulty configurations $q(k)$ decreases. Consequently, if faults persist, the system will eventually reach a state where no other faulty configurations are possible, i.e., the worst faulty condition. Thus, the number of potential switches is finite. Considering that the trajectory of an LTI switched system with stable modes cannot escape to infinity in finite time [105], [106], it is possible to conclude that system (4.50) will ultimately stabilize on a stable mode that may represent an intermediate faulty configuration or the worst faulty feasible configuration without loss of observability.

The same procedure can be followed for $\hat{e}(k)$. The evolution of the state is

$$x^n(k+1) = Ax^n(k) \tag{4.52}$$
$$y^n(k+1) = C(\gamma(k))Ax^n(k) + v(k+1), \tag{4.53}$$

and the observer equations are

$$\hat{x}^n(k+1) = A\hat{x}^n(k) + K_{\sigma(k)}(y^n(k+1) - \hat{y}^n(k+1)) \tag{4.54}$$
$$\hat{y}^n(k+1) = \bar{C}_{\sigma(k)} A\hat{x}^n(k) + \hat{b}(k+1). \tag{4.55}$$

[3] $\otimes$ denotes the Kronecker product.

The estimation error is

$$\hat{e}(k+1) \coloneqq x^n(k+1) - \hat{x}^n(k+1) \tag{4.56}$$

$$= Ae(k) + \bar{K}_{\sigma(k)} y^n(k+1) - K_{\sigma(k)} \bar{C}_{\sigma(k)} A\hat{x}^n(k) + \bar{K}_{\sigma(k)} \hat{b}(k+1) \tag{4.57}$$

$$= Ae(k) + \bar{K}_{\sigma(k)} C(\gamma(k)) Ax^n(k) - \bar{K}_{\sigma(k)} \bar{C}_{\sigma(k)} A\hat{x}_n(k) + \bar{K}_{\sigma(k)} v(k+1) + \bar{K}_{\sigma(k)} (b(k+1) - \hat{b}(k+1)). \tag{4.58}$$

Considering that

$$C(\gamma(k)) = C(\gamma(k)) + \bar{C}_{\sigma(k)} - \bar{C}_{\sigma(k)} \tag{4.59}$$

$$= \bar{C}_{\sigma(k)} + (C(\gamma(k)) - \bar{C}_{\sigma(k)}) \tag{4.60}$$

$$= \bar{C}_{\sigma(k)} + d_q(k). \tag{4.61}$$

Substituting (4.61) in (4.58) the error equation becomes

$$\hat{e}(k+1) = (A - \bar{K}_{\sigma(k)} \bar{C}_{\sigma(k)} A)\hat{e}(k) + d_2(k+1), \tag{4.62}$$

where $d_2(k)$ is

$$d_2(k+1) \coloneqq \bar{K}_{\sigma(k)} v(k+1) + \bar{K}_{\sigma(k)} d_q(k) Ax^n(k) + \bar{K}_{\sigma(k)} (b(k+1) - \hat{b}(k+1)), \tag{4.63}$$

that is bounded if the bias estimation error is bounded. Every mode of the switching system (4.62) is stable. Therefore, noting that (4.62) and (4.50) share the same switching signal $\sigma(k)$, the considerations discussed previously applies to (4.62). As a consequence, the switching systems (4.62) and (4.50) are stable concluding the proof. ■

It is worth remarking that global exponential stability for a switching system can be inferred by the asymptotic stability of each system only for "slow" enough switching signals $\sigma(k)$ [105]–[107]. As a consequence, if faults do not occur slowly over time the estimation performance will inevitably deteriorate (without compromising the stability of the strategy) until the overall network does set on a faulty condition.

4.4 Simulative Results

This concluding Section provides two numerical examples to illustrate the efficacy of the Sensor Reconciliation Architecture. The initial example employs a simplified model to illuminate interesting features of the architecture, whereas the second example utilizes the compartmental traffic model introduced in Chapter 3 for a more realistic application.

4.4.1 First Scenario

A scenario where five sensors work together to estimate the state of a plant is considered. The system parameters are:

$$A = \begin{bmatrix} 0.4 & 0.6 \\ 0.3 & 0.3 \end{bmatrix}, C^h = \begin{bmatrix} 1 & 0 & 1 & 0 & -1 \\ 0 & 1 & 1 & 2 & 0 \end{bmatrix}^T, \tag{4.64}$$

$$t_c = 0.1s, B = I_{n_x}, D = 0_{n_x}, \tag{4.65}$$

$$Q_v = 1 \times 10^{-3} I_m, N = 5, \tag{4.66}$$

where $C^h = C(\mathbf{1}_m)$ is the healthy output matrix and t_c is the sample time. Two exogenous inputs $u_1(k) = sin(k), u_2(k) = cos(k)$ are considered. In this example multiple faults occur during the simulation:

- Degradation of the effectiveness of sensors 1 and 2 leading to the complete loss of both sensors;
- Drop of the effectiveness of sensor 3 to the value 0.5;
- Additive faults (bias) on sensors 1 and 3 with

$$b_1(k) = sin\left(2\pi k/T\right), b_3(k) = sin\left(2\pi k/T\right), \tag{4.67}$$

  where $T = 16sec$ is the simulation time.

The topology of the network of sensors is a ring with weight 1 for each edge. Finally, a quantization step $\rho_q = 0.3$ is selected. Figure 4.2 illustrates the estimation process without the sensor reconciliation architecture. After the occurrence of the first faults at $k = 4s$, the estimation procedure becomes ineffective and the agents are unable to accurately reconstruct the state evolution. In contrast, when the simulation is repeated with the sensor reconciliation units enabled, the estimation performance is improved and the architecture is able to compensate both multiplicative and additive faults. More in details, the effectiveness of sensors 1 and 2 start decreasing at $k = 4s$ leading to the complete loss of both sensors at $k = 15s$. On the other hand, the effectiveness of sensor 3 drops abruptly to 0.5 at $k = 8s$. All DSR units are able to reconfigure themselves in order to better adapt to the new faulty configurations. A graphic evolution of the effectiveness parameters and their quantized values is provided in Figure 4.4. Moreover, the bias signals $b_1(k)$ and $b_3(k)$ appear on the outputs of sensors 1 and 3 at $k = 6s$. As shown in Figure 4.5, they are correctly estimated and compensated preserving the capability of the nodes to reach consensus and successfully estimate the state. It is worth noting that as effectiveness drops rapidly, the estimation of the bias becomes less accurate. This is evident when comparing the estimated values of $b_1(k)$ and $b_3(k)$ where sensor 1 experiences a gradual decrease in effectiveness over time, whereas sensor 3 encounters a sudden drop.

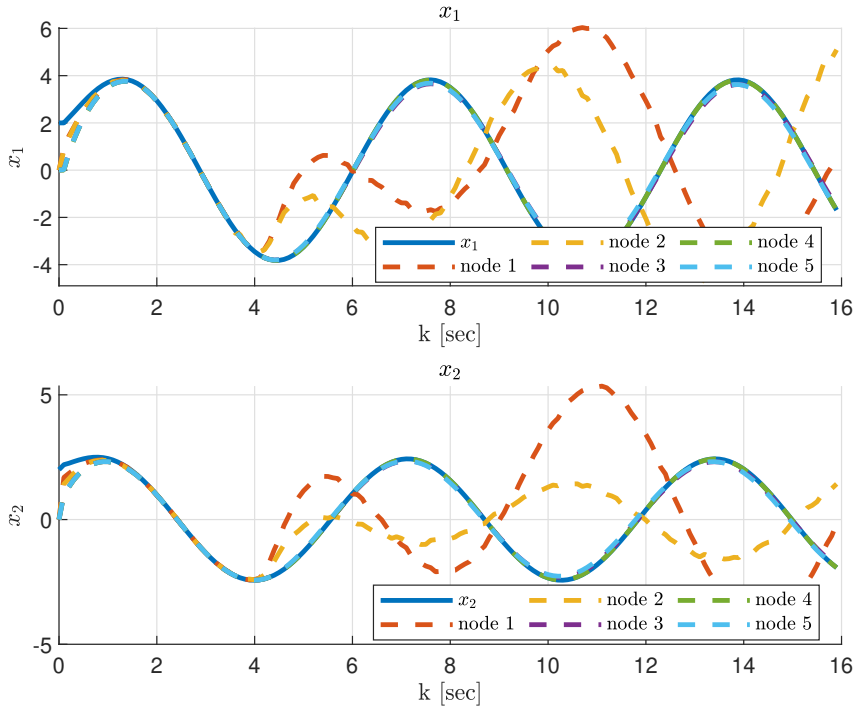

FIGURE 4.2: State estimation with DSR architecture disabled.

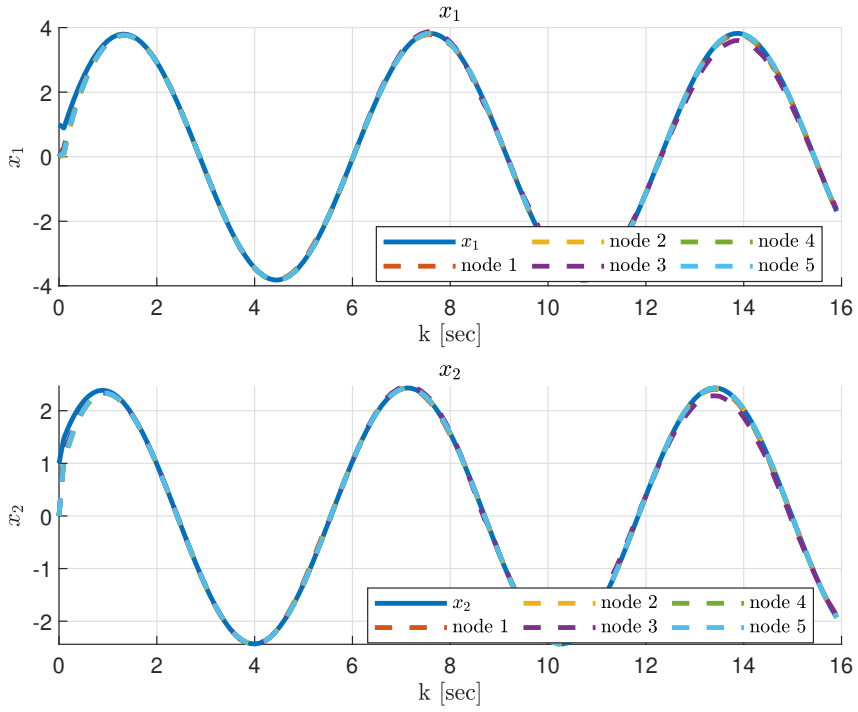

FIGURE 4.3: State estimation with DSR architecture enabled.

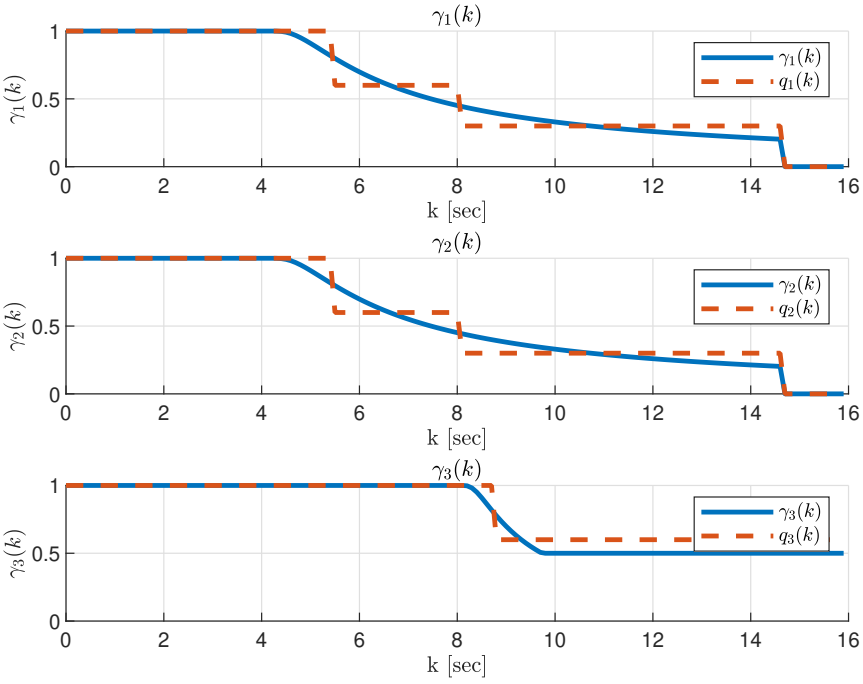

FIGURE 4.4: Evolution of the effectiveness parameters and their quantized values.

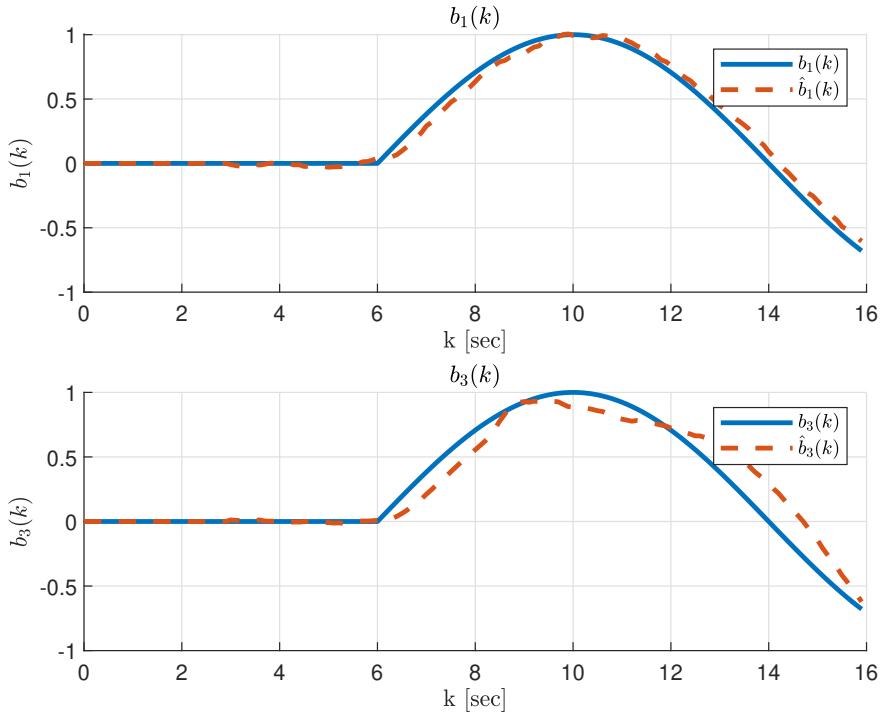

FIGURE 4.5: Estimation of the bias signals.

4.4.2 Second Scenario

In this final example, the performance of the distributed architecture is investigated in a more realistic setting. The compartmental model, previously introduced in Chapter 3 as a testbed, is monitored by a network of $N = 6$ sensors. Their spatial distribution is illustrated in Figure 4.6. This system represents a road network composed of $n_x = 21$ compartments.

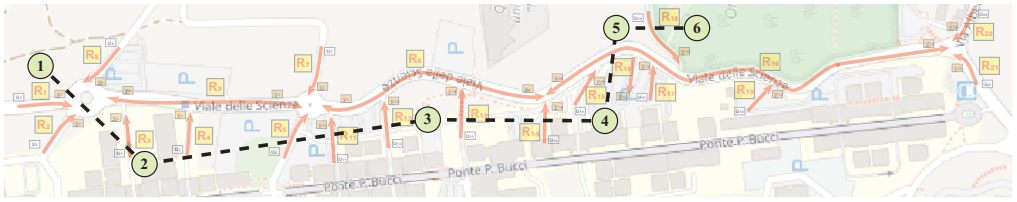

FIGURE 4.6: Location and interconnection of the nodes used for distributed traffic monitoring.

In Figure 4.6, the monitoring nodes are represented as green circles, while communication links are illustrated as black dashed lines. In this simulation, each sensor is assigned to a single node with the following sensor-node arrangement:

- Node 1 monitors section 1;
- Node 2 monitors section 3;
- Node 3 monitors section 12;
- Node 4 monitors section 15;
- Node 5 monitors section 16
- Node 6 monitors section 18.

Based on this information, the healthy output matrix is

$$C^h = \begin{bmatrix} (e_1^{n_x})^T \\ (e_3^{n_x})^T \\ (e_{12}^{n_x})^T \\ (e_{15}^{n_x})^T \\ (e_{16}^{n_x})^T \\ (e_{18}^{n_x})^T \end{bmatrix}. \tag{4.68}$$

To maintain realism, the communication network is defined based on geographical node locations, avoiding long-distance links. As a consequence, the resulting communication graph is a list. Sensors have been selected in order to guarantee observability and some degree of analytical redundancy. The sampling time is $tc = 1min$, the quantization step is $\rho_q = 0.3$ and the overall simulation time is $T = 180min$.

The following faults occurs during the simulation:

- Degradation of the effectiveness of sensors 1 and 2 leading to the complete loss of both sensors;
- Drop of the effectiveness of sensor 3 to the value 0.5;
- Additive faults (bias) on sensors 2 and 3 with

$$b_2(k) = sin\left(2\pi k/T\right), b_3(k) = sin\left(2\pi k/T\right). \tag{4.69}$$

Figure 4.7 shows the evolution of the effectiveness parameters for sensors associated to nodes 1, 2, 3 that monitor states 1, 3 and 12 respectively. The

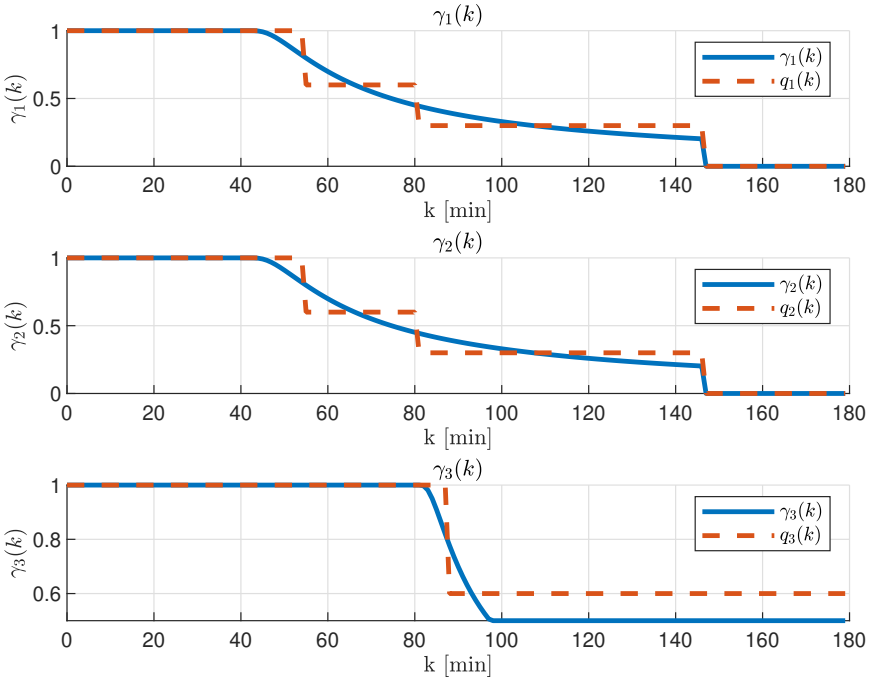

FIGURE 4.7: Evolution of the effectiveness parameters.

progression of multiplicative faults exhibits a pattern similar to the preceding example. In this instance, nodes 1 and 2 experience a complete loss of sensing capability after 150 minutes, whereas the efficacy of sensor 3 diminishes to half its original level.

As shown in Figure 4.8, the bias signals for sensors 2 and 3 are accurately estimated. Interestingly, the sudden decrease in sensor 3's effectiveness also impacts the bias estimation for node 3 without compromising the overall distributed estimation performance. In conclusion, Figure 4.9 shows the evolution of the states component x_1, x_2 and x_3 to graphically evaluate the estimation performance. All the nodes are able to estimate the evolution of the entire state despite the occurrences of unpredictable sensor faults. For a more comprehensive analysis

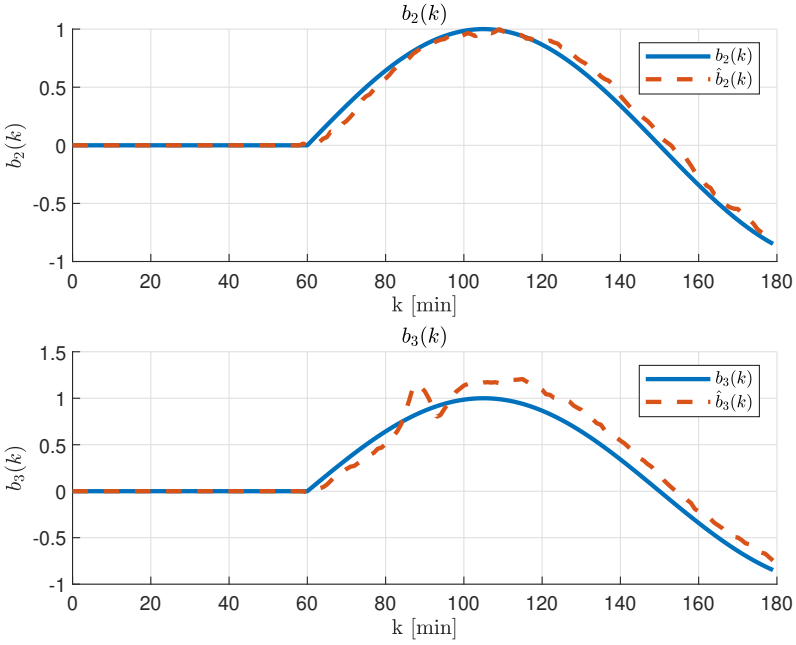

FIGURE 4.8: Estimation of the bias signals.

Table 4.1 reports the mean square errors associated to the estimation performed by each node for each state component.

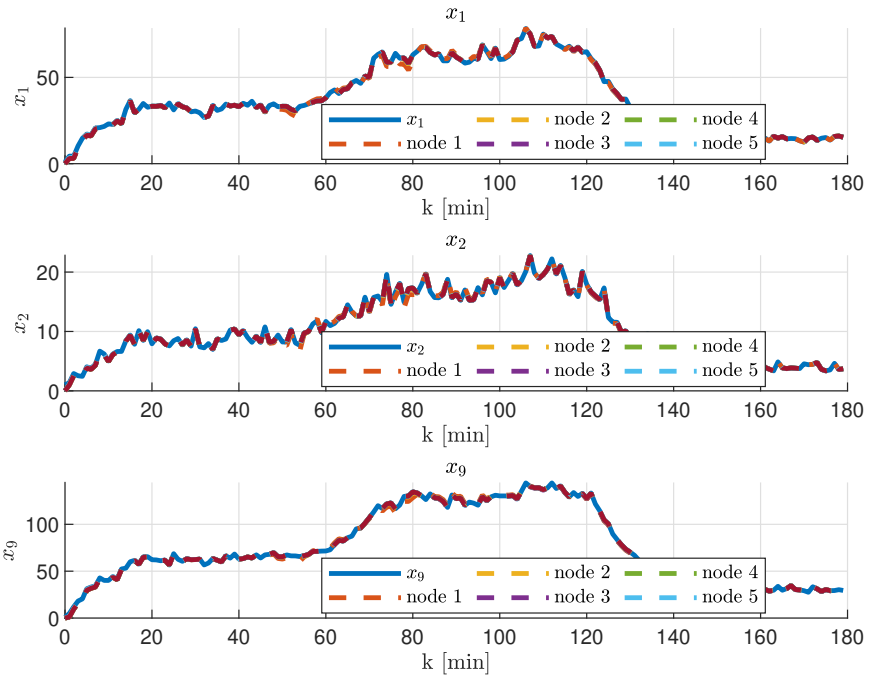

FIGURE 4.9: Evolution of three state components and their respective estimated values.

State \ **Node**	1	2	3	4	5	6
x_1	1.018	0.080	0.063	0.048	0.016	0.045
x_2	0.078	0.011	0.010	0.009	0.006	0.009
x_3	0.008	0.007	0.008	0.008	0.008	0.007
x_4	0.079	0.019	0.020	0.017	0.012	0.016
x_5	0.349	0.061	0.084	0.046	0.020	0.044
x_6	0.713	0.212	0.418	0.157	0.066	0.152
x_7	0.036	0.010	0.012	0.009	0.007	0.009
x_8	0.350	0.033	0.028	0.022	0.010	0.021
x_9	2.228	0.304	0.325	0.233	0.108	0.212
x_{10}	0.428	0.399	0.517	0.261	0.130	0.282
x_{11}	0.142	0.028	0.037	0.022	0.011	0.021
x_{12}	0.021	0.014	0.051	0.012	0.009	0.011
x_{13}	0.018	0.009	0.014	0.008	0.006	0.008
x_{14}	0.023	0.017	0.024	0.013	0.008	0.013
x_{15}	0.014	0.014	0.015	0.010	0.009	0.012
x_{16}	0.014	0.013	0.014	0.010	0.006	0.010
x_{17}	0.013	0.013	0.014	0.010	0.008	0.011
x_{18}	0.007	0.007	0.007	0.006	0.006	0.006
x_{19}	0.007	0.007	0.007	0.006	0.006	0.006
x_{20}	0.006	0.006	0.006	0.006	0.006	0.006
x_{21}	0.044	0.043	0.056	0.026	0.012	0.029

Table 4.1: Mean square errors for each sensor and for each state.

4.5 Conclusions

This Chapter addressed the distributed state estimation problem by implementing a Distributed Sensor Reconciliation architecture, which relied on steady-state Kalman filter as its core. This approach facilitated the full reconstruction of the central system's state, even when multiplicative and unexpected additive sensor faults were present. Additionally, the stability of the local estimators was examined and verified. To demonstrate the effectiveness of the proposed scheme, two numerical example involving two different models have been presented. In particular, the application of the distributed sensor reconciliation to a realistic traffic flow monitoring problem was discussed. Moving forward, future work will concentrate on the distributed estimation of the multiplicative fault parameters. This remains a crucial point to make the proposed architecture independent from external modules.

Sensor Selection and Observer Design Strategy for Positive Systems 5

This Chapter tackles the sensor selection problem, proposing a joint strategy to simultaneously select sensors and design an optimal L_1 observer. Specifically, the focus is on positive systems, a class of systems that presents unique challenges, necessitating the design of a positive observer to ensure positive estimated values. The Chapter is structured as follows: Section 5.1 introduces the sensor selection problem for positive systems; Section 5.2 presents the proposed solution and its theoretical guarantees; Section 5.3 demonstrates the strategy's effectiveness through numerical simulations; and finally, Section 5.4 offers concluding remarks.

5.1 Problem description

Consider the following discrete-time LTI state-space realization

$$\Sigma : \begin{cases} x(k+1) = Ax(k) + Bu(k) + B_w w(k) \\ \quad y(k) = Cx(k) + Du(k) + D_w w(k), \end{cases} \tag{5.1}$$

where: $k \in \mathbb{N}$, A, B, C, D, B_w, D_w are matrices of appropriate dimensions, $x(k) \in \mathbb{R}^{n_x}$ is the system state, $u(k) \in \mathbb{R}^{n_u}$ the system input, $w(k) \in \mathbb{R}^{n_w}$ the disturbance and $y(k) \in \mathbb{R}^m$ the measured output. It is worth recalling that the following statements about positive discrete-time systems (5.1) are equivalent [108]:

1. System (5.1) is said to be positive if for all $u(k) \geq 0$, $w(k) \geq 0$, $x(0) \geq 0$ the system state $x(k) \geq 0$ and the system output $y(k) \geq 0$, $\forall k \geq 0$;
2. System (5.1) is positive if and only if all entries of A, B, C, D, B_w, D_w are non-negative.

With reference to the sensor selection problem, in order to allow one to conveniently select a particular sensor from the set of all available sensors, the following vector of binary variables is introduced

$$h = [h_1, \ldots, h_m]^T \in \{0, 1\}^m, \tag{5.2}$$

where the variable h_i, $i = 1, \ldots, m$, indicates if the i-th sensor is selected or not. In particular, $h_i = 1$ means that the i-th sensor is used for measurement

whilst $h_i = 0$ means that the i-th sensor is excluded. Moreover, the quantity $h_a = \{h_{i_1}, \dots, h_{i_r}\}$ is used to represent the sub-set of $r \leq m$ sensors that have been selected and are active ($h_{i_1} = \cdots = h_{i_r} = 1$). Then, a sensor selection matrix $H(h_a)$ can be obtained from the diagonal matrix

$$H(h) := diag\,(h_1, \dots, h_m) \tag{5.3}$$

by eliminating the rows corresponding to non-active sensors. Specifically, $H(h_a)$ is a binary $r \times m$ matrix with $r \leq m$.

Then, the quantity $H(h_a)$ can be used in the state-space representation 5.1) for indicating the sensors that have been selected and are usable for measurement. This brings to the definition of the following system

$$\Sigma^+ : \begin{cases} x(k+1) = Ax(k) + Bu(k) + B_w w(k), \\ y(k) = Cx(k) + Du(k) + D_w w(k), \\ y_{h_a}(k) = H(h_a) y(k), \\ z_p(k) = C_z x(k), \end{cases} \tag{5.4}$$

with $y_{h_a}(k) \in \mathbb{R}^r$ that identify the system output that corresponds to the active sensors. An auxiliary variable $z_p(k)$ was also defined by introducing an additional output matrix C_z of arbitrary dimensions. This variable serves as a performance metric for observer design.

Given system (5.4), the joint sensor selection and observer design problem can be solved via an optimization problem whose aims can be summarized as follows:

- use the smallest possible number of sensors;
- design a suitable positive L_1 optimal Luenberger observer for full state reconstruction from inputs and active sensor measurements only;
- trade-off between number of selected sensors and the state reconstruction error with respect to a specific reconstruction metric;

Moving from these premise, for any choice h_a, the interest is on designing a positive L_1 optimal Luenberger observer of the form

$$O^+ : \begin{cases} \hat{x}(k+1) = A\hat{x}, (k) + Bu(k) + K(y_a(k) - \hat{y}_a(k)), \\ \hat{y}_a(k) = H(h_a)(C\hat{x}(k) + Du(k)), \\ \hat{z}_p(k) = C_z \hat{x}(k), \end{cases} \tag{5.5}$$

where $K \in \mathbb{R}^{n_x \times r}$ is the observer gain to be determined. By defining the state reconstruction error as $e(k) = x(k) - \hat{x}(k)$ and introducing the performance index $\tilde{z}_p(k) = z_p(k) - \hat{z}_p(k)$, the state reconstruction error dynamics is governed by the following equations:

$$\begin{aligned} e(k+1) &= (A - KH(h_a)C)e(k) + (B_w - KH(h_a)D_w)w(k), \\ \tilde{z}_p(k) &= C_z e(k), \end{aligned} \tag{5.6}$$

with K determined by minimizing the ∞-norm of $\tilde{z}_p(k)$. Notice that if $C_z = I_{n_x}$, the observer will be designed to filter $w(k)$ from the full state $x(k)$.

With all these elements, it is possible to state the joint sensor selection and observer design problem of interest as:

Problem 5.1.1. *(**Joint Sensor Selection and Observer Design**): Find a minimal number of sensors and a corresponding observer gain K for (5.5) such that:*

- *the observer (5.6) is positive, i.e. the matrices $(A - KH(h_a)C)$, $(B_w - KH(h_a)D_w)$ and C_z are non-negative for the selected sensors h_a;*
- *the state reconstruction error dynamics (5.6) is asymptotically stable;*
- *the observer gain minimizes the following L_1 performance index*

$$J := \sup_{w \in l_\infty} \frac{\|\tilde{z}_p\|_\infty}{\|w\|_\infty}, \tag{5.7}$$

where l_∞ denotes the vector space of bounded vector-valued sequences with $\|w\|_\infty = \sup_k \sum_i |w_i(k)|$, being $w_i(k)$ the i-th component of the vector $w(k)$.

It is important to point out that through this Chapter the inequalities are meant to be component wise and the notations $\mathbb{R}_+$ and $\mathbb{R}_{++}$ identify the set of non negative and positive real numbers respectively.

5.2 Joint Sensor Selection and Observer Design Problem

Before proceeding to discuss of an optimization procedure for Problem 5.1.1, the existence conditions of a solution are rigorously analyzed.

Proposition 5.2.1. *(L_1 **Optimal Positive Observer Characterization**): Given the state-space representation (5.4), the observer (5.5), the state reconstruction error dynamics (5.6) and a sensor selection h_a, the following statements are equivalent:*

(i) System (5.6) is positive and asymptotically stable and an observer gain K exists such that

$$\sup_{w \in l_\infty} \frac{\|\tilde{z}_p\|_\infty}{\|w\|_\infty} < \alpha, \quad \alpha \geq 0;$$

(ii) There exists a copositive matrix $P \in \mathbb{R}^{n_x}_{++}$ and a positive scalar $\alpha \in \mathbb{R}_+$ such that the following component wise inequalities

$$\begin{array}{rcl} P^T(A_{cl} - I) + \mathbf{1}_m^T C_z & < & 0 \\ P^T B_{cl} - \alpha \mathbf{1}_{n_x}^T & < & 0, \end{array} \tag{5.8}$$

where $A_{cl} = A - KH(h_a)C$ and $B_{cl} = B_w - KH(h_a)D_w$, are feasible and such that the following component wise inequalities

$$\begin{array}{rcl} P^T(A - LH(h_a)C) & \geq & 0 \\ P^T(B_w - LH(h_a)D_w) & \geq & 0, \end{array} \tag{5.9}$$

holds true. Moreover, under a positive observer (5.5), the error dynamics (5.6) is positive.

Proof. The proof relies on the dissipativity theory for non-negative systems [109]. Consider the discrete-time state-space representation (5.6), the copositive linear storage function $V(e(k)) = P^T e(k)$ with $P \in \mathbb{R}^{n_x}_{++}$ and the dissipation inequality

$$V(e(k+1)) - V(e(k)) \leq s(w(k), \tilde{z}_p(k)), \forall k \in \mathbb{N}, \tag{5.10}$$

where:

- $V : \mathbb{R}^{n_x} \to \mathbb{R}$ is a storage function that generalizes the energy function for a dissipative system;
- $s : \mathbb{R}^{n_w} \times \mathbb{R}^m \to \mathbb{R}$ is a supply function representing the rate at which the system absorbs energy.

Then, according to dissipativity theory, if the left side of (5.10) decreases along the trajectories of the system (5.6), then the system is dissipative with respect to the supply rate $s(w(k), \tilde{z}_p(k))$, and the L_1-gain is bounded from above by α. Furthermore, because $P > 0$, the asymptotic stability of the system also follows. Starting from the definition of the L_1 norm it is possible to write

$$\sup_{w \in l_\infty} \frac{\|\tilde{z}_p(k)\|_\infty}{\|w(k)\|_\infty} < \alpha \to \|\tilde{z}_p(k)\|_\infty < \alpha \|w(k)\|_\infty \, \forall w(\cdot) \in l_\infty. \tag{5.11}$$

The supply function $s(w(k), \tilde{z}(k)) < 0$, $\forall k$, can be defined as

$$\begin{array}{rl} s(w(k), \tilde{z}_p(k)) = \alpha \|w(k)\|_\infty - \|\tilde{z}_p(k)\|_\infty = & \\ = \alpha \mathbf{1}_{n_x}^T w(k) - \mathbf{1}_m^T \tilde{z}_p(k), & \alpha > 0, \end{array} \tag{5.12}$$

that, by accounting for the state-space representation (5.6), can be rewritten as

$$s(w(k), C_z e(k)) = \alpha \mathbf{1}_{n_x}^T w(k) - \mathbf{1}_m^T \left[C_z e(k)\right]. \tag{5.13}$$

The storage function $V(e(k)) = P^T e(k)$ that is related to the supply function $s(w(k), \tilde{z}_p(k))$ can be viewed as a Lyapunov candidate function. Consequently, it is possible to compute

$$\begin{array}{rcl} \Delta V(e(k)) & = & V(e(k+1)) - V(e(k)) \\ & = & P^T e(k+1) - P^T e(k). \end{array} \tag{5.14}$$

Then, by replacing (5.6) in (5.14), it is possible to obtain

$$\begin{array}{rcl} \Delta V(e(k)) & = & \left[P^T(A - KH(h_a)C) - P^T\right] e(k) + \\ & & + P^T(B_w - KH(h_a)D_w) w(k). \end{array} \tag{5.15}$$

By substituting eqs. (5.15) and (5.13) in (5.10), the following component wise inequality is obtained

$$\begin{bmatrix} P^T(A - LH(h_a)C) - P^T + \mathbf{1}_m^T C_z \\ P^T(B_w - LH(h_a)D_w) - \alpha \mathbf{1}_{n_x}^T \end{bmatrix}^T \begin{bmatrix} e(k) \\ w(k) \end{bmatrix} \leq 0, \tag{5.16}$$

that finally becomes

$$\begin{bmatrix} P^T(A_{cl} - I) + \mathbf{1}_m^T C_z \\ P^T(E_{cl} - I) - \alpha \mathbf{1}_{n_x}^T \end{bmatrix}^T \begin{bmatrix} e(k) \\ w(k) \end{bmatrix} \leq 0 \tag{5.17}$$

by denoting $A_{cl} = A - KH(h_a)C$ and $B_{cl} = B_w - KH(h_a)D_w$. Because sequences $e(k)$ and $w(k)$ are non-negative, then $\Delta V(e(k))$ is negative on $\mathbb{R}_+^{n_x}$ if and only if the left factor in (5.17) is negative or equivalently if the conditions (5.8) hold true. Furthermore, since $P > 0$ and in accordance with eq. (5.9), it follows that $A_{cl} \geq 0$ and $E_{cl} \geq 0$. ■

Proposition 5.2.2. *(**Sensors Selection and Observer Design**): With reference to sensors selection and L_1 optimal positive observer design problem, the observer gain, if it exists, can be computed as $L = X^{*-1}U^*$ by solving the following multi-objective mixed-integer semi-definite optimization problem:*

$$[X^*, U^*, h_a^*, \alpha^*] = arg \min_{X,\ U,\ h,\ \alpha} \mathbf{a_1} \cdot \mathbf{c}^T h + \mathbf{a_2}\alpha \tag{5.18}$$

$$\begin{bmatrix} \mathbf{1}_{n_x}^T \\ \mathbf{1}_m^T \end{bmatrix}^T \begin{bmatrix} X(A - I) - UC & XB_w - UD_w - \alpha \mathbf{1}_{n_x}^T \\ C_z & 0 \end{bmatrix} \leq 0, \tag{5.19}$$

$$X \in \mathbb{R}^{n_x \times n_x} \text{ is diagonal}, \quad X\mathbf{1}_{n_x} > 0, \tag{5.20}$$

$$\alpha > 0, \quad h = \{0, 1\}^m, \tag{5.21}$$

$$\Phi\nu_1(U, Y, h) \leq \nu_2, \tag{5.22}$$

$$\sum_{i=1}^{m} h_i \geq \underline{r}. \tag{5.23}$$

Proof. It follows from proof of Proposition 5.2.1. Note that, for sensors selection and observer design purpose, in what follows $H(h_a)$ is replaced by $H(h)$ and consequently inequality (5.16) can be rewritten as

$$\begin{bmatrix} P^T A - P^T KH(h)C - P^T + \mathbf{1}_m^T C_z \\ P^T B_w - P^T KH(h)D_w - \alpha \mathbf{1}_{n_x}^T \end{bmatrix}^T \begin{bmatrix} e(k) \\ w(k) \end{bmatrix} \leq 0. \tag{5.24}$$

By performing the change of variables $\mathbf{1}_{n_x}^T X = P^T$, with $X \in \mathbb{R}^{n_x \times n_x}$ diagonal, and $Y = XK$ one gets

$$\begin{bmatrix} \mathbf{1}_{n_x}^T XA - \mathbf{1}_{n_x}^T YH(h)C - \mathbf{1}_{n_x}^T X + \mathbf{1}_m^T C_z \\ \mathbf{1}_{n_x}^T XB_w - \mathbf{1}_{n_x}^T YH(h)D_w - \alpha \mathbf{1}_n^T \end{bmatrix}^T \begin{bmatrix} e(k) \\ w(k) \end{bmatrix} \leq 0. \tag{5.25}$$

Notice that term $YH(h)$ in (5.25) is nonlinear being the product of the slack variables Y and $H(h)$. However, by applying the McCormick relaxation procedure

[110], it is possible to relax condition (5.25) into a linear inequality. This reformulation can be achieved by introducing the new matrix variable $U \coloneqq YH(h)$. Specifically, assume that Y could be bounded so that $\underline{Y} \leq Y \leq \overline{Y}$. Then, the McCormick's relaxation allows one to write the inequalities (5.25) as

$$\underbrace{\begin{bmatrix} \mathbf{1}_{n_x}^T XA - \mathbf{1}_{n_x}^T UC - \mathbf{1}_{n_x}^T X + \mathbf{1}_m^T C_z \\ \mathbf{1}_{n_x}^T XE - \mathbf{1}_{n_x}^T UF - \alpha \mathbf{1}_{n_x}^T \end{bmatrix}^T}_{M} \begin{bmatrix} e(k) \\ w(k) \end{bmatrix} \leq 0. \tag{5.26}$$

From inequality (5.26), it is evident that $M < 0$, in a component wise fashion, because $e(k)$ and $w(k)$ are non-negative sequences. Moreover, M can be written again as

$$\begin{bmatrix} \mathbf{1}_{n_x}^T \\ \mathbf{1}_m^T \end{bmatrix}^T \begin{bmatrix} X(A-I) - UC & XB_w - UD_w - \alpha \mathbf{1}_n^T \\ C_z & 0 \end{bmatrix} \leq 0, \tag{5.27}$$

$$U = YH(h), \quad \underline{Y} \leq Y \leq \overline{Y}. \tag{5.28}$$

Following the same procedure reported in [111], conditions (5.28) can be rewritten as

$$\Phi \nu_1(U, Y, h) \leq \nu_2. \tag{5.29}$$

Now, by focusing on conditions (5.28), it can be observed that $U = YH(h)$ is equivalent to $U_{(i,j)} = Y_{(i,j)}h$ for all i, j since $H(h)$ is a diagonal matrix. Then, the following equivalence holds

$$U_{(i,j)} = Y_{(i,j)}h_j \Leftrightarrow U_{(i,j)} = \begin{cases} Y_{(i,j)}, & if \quad h_j = 1, \\ 0, & if \quad h_j = 0, \end{cases} \tag{5.30}$$

where $Y_{(i,j)} \in \left[\underline{Y}_{(i,j)}, \overline{Y}_{(i,j)}\right]$ for all i, j. The transformation of (5.30) into a mixed-integer semi-definite optimization problems is carried out through the application of the McCormick's relaxation, the rationale of which can be elucidated as follows. First of all, it must be highlighted that all terms $(\overline{Y}_{(i,j)} - Y_{(i,j)})$, $(Y_{(i,j)} - \underline{Y}_{(i,j)})$, $(1 - h_j)$ and h_j are nonnegative. Moreover, by considering that $U_{(i,j)} = Y_{(i,j)}h_j$, the following inequalities straightforwardly results

$$\begin{cases} (\overline{Y}_{(i,j)} - Y_{(i,j)})(1 - h_j) \geq 0, \\ \qquad\qquad \Leftrightarrow U_{(i,j)} \geq Y_{(i,j)} + \overline{Y}_{(i,j)}(h_j - 1), \\ \qquad (\overline{Y}_{(i,j)} - Y_{(i,j)})h_j \geq 0 \Leftrightarrow \overline{Y}_{(i,j)}h_j \geq U_{(i,j)}, \\ (Y_{(i,j)} - \underline{Y}_{(i,j)})(1 - h_j) \geq 0, \\ \qquad\qquad \Leftrightarrow -U_{(i,j)} \geq -Y_{(i,j)} + \underline{Y}_{(i,j)}(1 - h_j), \\ \qquad (Y_{(i,j)} - \underline{Y}_{(i,j)})h_j \geq 0 \Leftrightarrow U_{(i,j)} \geq \underline{Y}_{(i,j)}h_j. \end{cases} \tag{5.31}$$

Then, by noting that $h_j \in \{0, 1\}$, when $h_j = 1$ is used in (5.31), one has

$$\left.\begin{array}{lll} Y_{(i,j)} \geq & U_{(i,j)} & \geq Y_{(i,j)} \\ \overline{Y}_{(i,j)} \geq & U_{(i,j)} & \geq \underline{Y}_{(i,j)} \end{array}\right\} \Rightarrow U_{(i,j)} = Y_{(i,j)}, \tag{5.32}$$

while, for $h_j = 0$, one obtains

$$\left.\begin{array}{rll} 0 \geq & U_{(i,j)} & \geq 0 \\ \overline{Y}_{(i,j)} \geq & Y_{(i,j)} & \geq \underline{Y}_{(i,j)} \end{array}\right\} \Rightarrow U_{(i,j)} = 0. \tag{5.33}$$

Because conditions (5.32)-(5.33) hold for all i, j, then (5.30) and (5.31) are equivalent. Now, it is possible to introduce the following terms

$$\sigma_1 = [1,\, -1,\, 1,\, -1]^T, \quad \sigma_2 = [-1,\, 1,\, 0,\, 0]^T, \tag{5.34}$$

$$\Psi = \Omega' \oslash [I_m \otimes \mathbf{1}_{4n_x}], \quad \Omega' = \underbrace{[vec(\Omega) \times vec(\Omega)]}_{m\,times}, \tag{5.35}$$

$$\Omega = \begin{bmatrix} \omega_{(1,1)} & \omega_{(1,2)} & \dots & \omega_{(1,m)} \\ \omega_{(2,1)} & \omega_{(2,2)} & \dots & \omega_{(2,m)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{(n_x,1)} & \omega_{(n_x,2)} & \dots & \omega(n_x, m) \end{bmatrix}, \omega_{(i,j)} \coloneqq \begin{bmatrix} -\underline{Y}_{(i,j)} \\ \overline{Y}_{(i,j)} \\ -\overline{Y}_{(i,j)} \\ \underline{Y}_{(i,j)}, \end{bmatrix}, \tag{5.36}$$

$$\psi = vec(\Omega) \oslash \left(\mathbf{1}_{n_x m} \otimes [1,\, 1,\, 0,\, 0]^T\right), \tag{5.37}$$

Where $\oslash$ is the Hadamard product and $\otimes$ is the Kronecker product. Then, for all i, j it is possible to rewrite (5.31) as

$$[I_{n_x m} \otimes \sigma_1 | I_{n_x m} \otimes \sigma_2 | \Psi] \left[vec(U)^T,\, vec(Y)^T,\, h^T\right]^T \leq \psi. \tag{5.38}$$

Finally, by combining (5.38) with $\underline{Y} \leq Y \leq \overline{Y}$, it is possible to obtain (5.22) where

$$\Phi \coloneqq blkdiag([I_{n_x m} \otimes \sigma_1 | I_{n_x m} \otimes \sigma_2 | \Psi], I_{n_x m}, -I_{n_x m}), \tag{5.39}$$

$$\nu_2 \coloneqq \left[\psi^T,\, vec(\overline{Y})^T,\, -vec(\underline{Y})^T\right], \tag{5.40}$$

$$\nu_1(U, Y, h) \coloneqq \begin{bmatrix} \left[vec(U)^T,\, vec(Y)^T,\, h^T\right]^T \\ vec(Y) \\ vec(Y) \end{bmatrix}. \tag{5.41}$$

■

Remark 5.2.1. *It should be emphasized that, provided system (5.1) is observable, the mixed integer problem formulated in Proposition 5.2.2 is always feasible. In fact, in the worst case scenario, it might be necessary to select m sensors, where m is the maximum number of sensors available. Notice also that it is feasible to trade off between the observer's optimality and the minimization of the number of activated sensors in the multi-objective index of (5.19)-(5.23). Specifically, to construct such a trade-off, the constants* $\mathbf{a_1} \in \mathbb{R}_+$ *and* $\mathbf{a_2} \in \mathbb{R}_+$ *are utilized. Moreover, each sensor* h_i *is given a weight via the vector* $\mathbf{c}^T$*, which can be utilized to add more preference knobs to further alter the solution. It is also important to note that weights* $\mathbf{a_1}$ *and* $\mathbf{a_2}$ *have a slight effect on the solution, and effective solutions can be obtained with a wide variety of values. All that matters is that the two terms in the cost must be different than zero. In fact, tests have suggested that solutions that guarantee Observability are obtained by using a set*

of sensors whose cardinality surpasses the minimum when $\mathbf{a_1}$ *is selected near to* 0. *Conversely, minimal sensor sets that cannot ensure Observability result from selecting* $\mathbf{a_2}$ *too small or equal to* 0. *Finally, a sensor budget constraint (5.23) is present for enforcing a lower bound* $\underline{r}$ *on the number of sensors that must be selected.*

Remark 5.2.2. *Although the constraints in Problem (5.19)-(5.23) are formulated as component-wise inequalities, they can be easily expressed in SDP form by defining a diagonal matrix where each diagonal element corresponds to the left-hand side of the each (component-wise) inequality. The component-wise constraints are then imposed as semi-definite conditions on the diagonal matrix [112].*

5.3 Numerical Simulations

In this Section the performance and efficacy of the proposed strategy are assessed by applying the joint sensor selection and observer design strategy to road network systems modeled as positive compartmental models.

5.3.1 First Scenario

In this initial illustrative example, the strategy is employed to construct a monitoring system for a compact road network.

$$A = \begin{pmatrix} 0.36154 & 0.27273 & 0 & 0 & 0 & 0 & 0 & 0.25 & 0.15 \\ 0.07692 & 0.081818 & 0 & 0 & 0 & 0 & 0 & 0.0833 & 0.05 \\ 0 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 0 & 0.3 & 0.1875 & 0.23077 & 0 & 0.15 \\ 0 & 0 & 0 & 0 & 0.26667 & 0.4 & 0.30769 & 0 & 0.2 \\ 0 & 0 & 0 & 0 & 0.06667 & 0.0625 & 0.053846 & 0 & 0.05 \\ 0.15385 & 0.18182 & 0 & 0 & 0 & 0 & 0 & 0.233 & 0.1 \\ 0.30769 & 0.36364 & 0.5 & 0.5 & 0.2667 & 0.25 & 0.30769 & 0.333 & 0.1 \end{pmatrix}; \quad (5.42)$$

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}; \quad (5.43)$$

Consider the discrete-time LTI model (5.1) characterized by the system matrices (5.42)-(5.43) and describing the road network depicted in Figure 5.1. In a similar way to that seen in Chapter 3, this network is modeled as a positive compartmental system, where the system state represents the traffic flows (vehicles per unit time) in specific sections of the road network. As shown in Figure 5.1, the compartments (or road sectors) are illustrated as boxes, with the flows

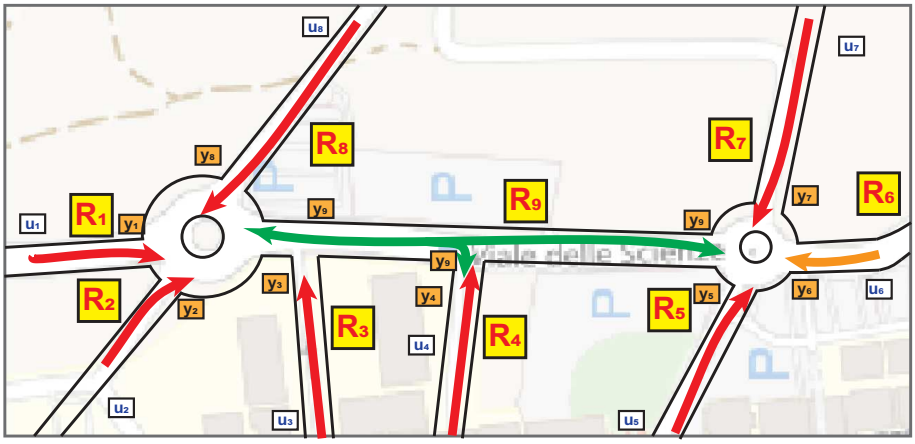

FIGURE 5.1: Case study - traffic network. R_i denote 9 potential sensor locations, u_i and y_i the vehicles entering and leaving the i-th road sector, respectively.

between them represented by arrows. Additionally, u_i and y_i denote the vehicles entering and leaving the i-th road sector, respectively. Note that in all tests, signals representing varying traffic conditions in the network were used as inputs. Additionally, the error signal $w(k)$ is used to introduce a sort of 'bias' error made by the sensors in measuring the correct vehicle flows. More in detail, $w(k)$ has been modeled as a pulse train with amplitude of 2, period of 10 seconds, and a duty cycle of 5%.

Test	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{c}$	C_z	$\mathcal{H}^{\mathbf{opt}}$	$\lvert\mathcal{H}^{\mathbf{opt}}\rvert$	$\alpha^{\mathbf{opt}}$
1	1	0.1	$\mathbf{c}_1^{S1}$	I_n	$[h_3, h_9]$	2	0.1394
2	0.1	1	$\mathbf{c}_1^{S1}$	I_n	$[h_4, h_9]$	2	0.1267
3	1	0.1	$\mathbf{c}_2^{S1}$	I_n	$[h_4]$	1	0.1267
4	0.01	1	$\mathbf{c}_2^{S1}$	I_n	$[h_3, h_4]$	2	0.1130
5	1	0.1	$\mathbf{c}_1^{S1}$	$H(h_a)C$	$[h_3, h_7]$	2	0.0225
6	0.1	1	$\mathbf{c}_1^{S1}$	$H(h_a)C$	$[h_3, h_7]$	2	0.0225
7	1	0.1	$\mathbf{c}_2^{S1}$	$H(h_a)C$	$[h_4]$	1	0.0121
8	0.01	1	$\mathbf{c}_2^{S1}$	$H(h_a)C$	$[h_4, h_7]$	2	0.0235

Table 5.1: Solution to the optimization problem (5.18)-(5.23): objective function (5.18) weighting parameters ($\mathbf{a_1}$, $\mathbf{a_2}$, $\mathbf{c}$), objective vector matrix (C_z), optimal sensor set ($\mathcal{H}^{\mathbf{opt}}$), optimal sensor set cardinality ($|\mathcal{H}^{\mathbf{opt}}|$), bound of the optimal L_1 norm ($\alpha^{\mathbf{opt}}$).

Several tests have been performed and the optimization problem (5.18)-(5.23) has been solved for different values of the weight parameters $\mathbf{a_1}$, $\mathbf{a_2}$ and $\mathbf{c}$. The choice of the parameters $\mathbf{a_1}$ and $\mathbf{a_2}$ allows one to put more emphasis on finding a solution with a minimum number of sensors than preferring to reduce the state reconstruction error and vice versa (e.g., choosing $\mathbf{a_1} > \mathbf{a_2}$ means that the focus is on putting more emphasis on determining the minimum number of sensors). In addition, the parameters vector $\mathbf{c}$ can be used to assign a "priority" to such

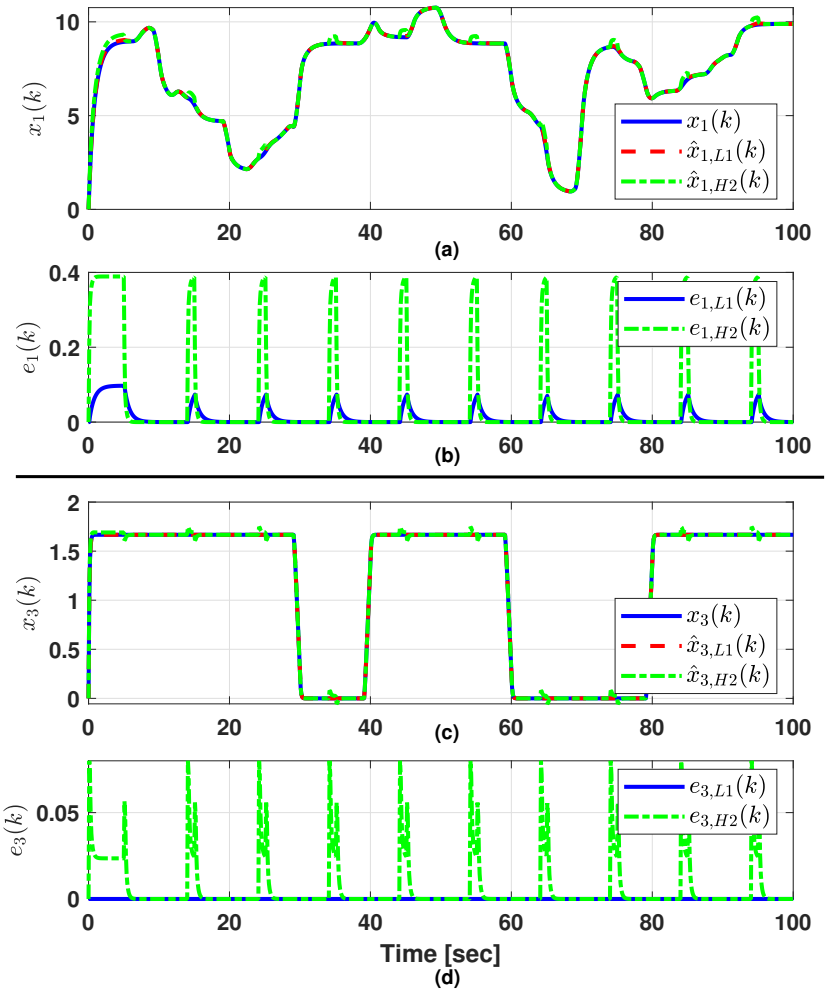

FIGURE 5.2: Figures (a) and (c): System state evolution ($x_1(k)$ and $x_3(k)$ - blue line) and system state estimates provided by the proposed L_1 observer ($\hat{x}_{1,L1}(k)$ and $\hat{x}_{3,L1}(k)$ - red dashed line) and H_2 observer ($\hat{x}_{1,H2}(k)$ and $\hat{x}_{3,H2}(k)$ - green line). Figures (b) and (d): state reconstruction error evolutions ($e_{1,L1}(k)$, $e_{1,H2}(k)$, $e_{3,L1}(k)$, $e_{3,H2}(k)$).

system output where it is desirable to allocate the sensors (e.g. for easiness or cost-effectiveness of the installation in environmental monitoring application). The eight tests were divided into two groups based on the definition of the performance output $z_p(k)$: the first four (Tests 1-4) employed $C_z = I_{n_x}$, while the last four (Tests 5-8) employed $C_z = H(h_a)C$. Table 5.1 shows the value of the parameters taken into account for the tests under consideration. Note that the tests have been performed by considering, alternatively, the following values

for the vector $\mathbf{c}$:

$$\begin{array}{rcl} \mathbf{c}_1^{S1} & = & \mathbf{1}_9, \\ \mathbf{c}_2^{S1} & = & [0.4\ 0.4\ 0.5\ 0.4\ 0.5\ 0.4\ 0.4\ 0.5\ 0.5]\,. \end{array} \tag{5.44}$$

Notice that $\mathbf{c}_2^{S1}$ assigns a different priority to each system output while $\mathbf{c}_1^{S1}$ provides an instance where the preference is uniform.

According to the results depicted in Figure 5.2 and shown into Table 5.1, it is possible to confirm that the procedure proposed in this study has a satisfactory performance both in terms of sensor selection and state reconstruction error.

The efficacy of the proposed method has also been assessed by comparisons with a H_2 Luenberger observer, especially with regard to system state reconstruction. In particular, the H_2 observers was built utilizing the available sensor set $\mathcal{H}^{\mathbf{opt}} = [h_4]$, taking into account the outcomes of Test 7. The outcomes are displayed in Figure 5.2. In particular, Figures 5.2(a)-(c) report a comparison between the system state evolution and system state estimates provided by the proposed L_1 observer and the H_2 observer. Moreover, Figures 5.2(b)-(d) report the evolution of the state reconstruction error, estimated by the proposed L_1 observer and the H_2 observer. The outcomes unequivocally show that, even with the identical sensor set $\mathcal{H}^{\mathbf{opt}}$, the suggested L_1 observer performs better than the H_2 observer in the scenario of interest. For the remaining tests listed in Table 5.1 - which are omitted for the purpose of conciseness - similar outcomes were obtained.

5.3.2 Second Scenario

The final simulative scenario is devoted to illustrate the use of the proposed methodology for solving the joint sensor selection and observer design problem for traffic monitoring in a realistic and wider urban road network. The road network model used for the simulations is the one discussed in Chapter 3 (composed of $n_x = 21$ compartments/states). Several tests have been considered with the aim at finding the minimum number of traffic sensors to be placed in a few specific road segment that still allows the correct full state reconstruction in all segments of the road network. The simulations have been undertaken via MATLAB R2022b running on a 64-bit Windows 10 with 3.5GHz Intel(R) Xeon(R) W-2265 CPU and 64 GB of RAM. The optimization problem (5.18)-(5.23) via the YALMIP's [113] BnB algorithm along with the MOSEK's [114] SDP solver. In this case as well, the optimization problem (5.18)-(5.23) was solved for various values of the weight parameters $\mathbf{a_1}$, $\mathbf{a_2}$, and $\mathbf{c}$. The introduction of the weight parameters is particularly useful when designing a urban smart architecture, for example, a smart lighting system that adjusts lamp brightness based on traffic information provided by sensors or cameras [115]. In fact, this design approach helps to trade off technological costs with performance criteria effectively. Table 5.2 shows the value of the parameters taken into account for the tests under consideration. Note that, the tests have been performed by accounting, alterna-

tively, the following values for the vector $\mathbf{c}$:

$$\begin{aligned} \mathbf{c}_1^{S2} &= [0.5\ 0.5\ 0.5\ 0.4\ 0.5\ 0.6\ 0.5\ 0.4\ 0.7\ 0.6 \\ &\quad 0.4\ 0.6\ 0.4\ 0.4\ 0.5\ 0.1\ 0.1\ 0.6\ 0.1\ 0.7\ 1.0]\,, \\ \mathbf{c}_2^{S2} &= \mathbf{1}_{21}. \end{aligned} \tag{5.45}$$

Furthermore, notice that $\mathbf{c}_1^{S2}$ assigns different priority to each road segment while, on the other hand, $\mathbf{c}_2^{S2}$ provides an instance where the preference is uniform. Eight tests were conducted, divided into two groups based on the definition of the performance output $z_p(k)$: the first four (Tests 1-4) used $C_z = I_{n_x}$, while the last four (Tests 5-8) employed $C_z = H(h_a)C$. This means that in Tests $1-4$ the aim was at finding a solution that guarantees homogeneous performance on all state components while in Tests $5-8$ one is more interested in ensuring good performance on the road segments where are allocated the sensors. Table 5.2 summarizes the results related to all tests. In details, it reports the weighting parameters ($\mathbf{a_1}$, $\mathbf{a_2}$, $\mathbf{c}$) used to solve the optimization problem (5.18)-(5.23), the chosen objective vector matrix (C_z), the computed optimal sensor set ($\mathcal{H}^{opt}$), the optimal sensor set cardinality ($|\mathcal{H}^{opt}|$), the achieved bound on the L_1 norm (α^{opt}) and the execution time (τ).

It is worth remarking that in all tests an additional error signal acting as "bias" sensor error $w(k)$ on the sensors has been considered. Note that, the following considerations can be made on $w(k)$:

- it represents the error accomplished by the sensors in determining the correct number of vehicles in the road sector;
- it has been modeled as a pulse train with amplitude equals to 2, period equals to 10 sec and duty cycle equals to $5\%w(k)$.

Further comparative studies have been conducted to evaluate the performance of the positive L_1 Luenberger observer (5.5), which was designed by solving the optimization problem (5.18)-(5.23), in terms of state error reconstruction. For this purpose, a conventional H_2 Luenberger observer has been designed as well on the basis of the same sets of sensors $\mathcal{H}^{opt}$ achieved as solutions in Tests 1-8 and used for comparisons. The same disturbance sequence $w(k)$ was used in the comparative tests.

Figures 5.3 - 5.4 report the system state (x_4 and x_{13}) evolutions, where the blue line represents the state, the red dashed line the state estimate $\hat{x}_4^{L1}$ provided by the proposed L_1 observer and the green dashed line the state estimate $\hat{x}_4^{H2}$ provided by the standard H_2 observer. Moreover, Figures 5.5 - 5.6 report the state reconstruction error evolutions ($e_4(k)$, $e_{13}(k)$) for $k \in [2900, 3000]$ [sec]. Note that in this latter figures, the blue line refers to the reconstruction error obtained by using the proposed L_1 positive observer while the green line refers to the reconstruction error obtained by using the standard H_2 observer. Furthermore, it is worth pointing out that Figures 5.3 - 5.6 refers to Test 1 (i.e. $\mathcal{H}^{opt} = [h_4, h_{13}, h_{16}, h_{17}, h_{19}]$). Comparable results were achieved for Tests 2-8, which are omitted here for the sake of brevity.

According to the results depicted in Figures 5.3-5.4 and those results shown into Table 5.2, where the last two rows (for each test) report the ∞-norms of

Test	1	2	3	4
$\mathbf{a_1}$	1	0.1	10^{-4}	1
$\mathbf{a_2}$	0.1	1	0.1	1
$\mathbf{c}$	$\mathbf{c}_1^{S2}$	$\mathbf{c}_1^{S2}$	$\mathbf{c}_2^{S2}$	$\mathbf{c}_2^{S2}$
$\mathbf{C_z}$	I_{n_x}	I_{n_x}	I_{n_x}	I_{n_x}
$\mathcal{H}^{\mathbf{opt}}$	$[h_4, h_{13}, h_{16}, h_{17}, h_{19}]$	$[h_4, h_{13}, h_{16}, h_{17}, h_{19}, h_{21}]$	$[h_3, h_4, h_{12}, h_{13}, h_{15}, h_{16}, h_{17}, h_{18}, h_{21}]$	$[h_1, h_3, h_5, h_6, h_7, h_{11}, h_{13}, h_{14}, h_{15}, h_{16}, h_{19}, h_{21}]$
$\lvert\mathcal{H}^{\mathbf{opt}}\rvert$	5	6	9	12
α^{opt}	0.4146	0.3287	0.0148	0.0140
τ	12.34[sec]	7.12[sec]	1.05[sec]	0.38[sec]
$\lVert\mathbf{e}\rVert_\infty^{\mathbf{L_1}}$	0.1762	0.0610	0.0680	0.0584
$\lVert\mathbf{e}\rVert_\infty^{\mathbf{H_2}}$	1.5155	1.1645	1.2602	0.9771
Test	**5**	**6**	**7**	**8**
$\mathbf{a_1}$	10	0.1	2	1
$\mathbf{a_2}$	0.1	10	1	2
$\mathbf{c}$	$\mathbf{c}_1^{S2}$	$\mathbf{c}_1^{S2}$	$\mathbf{c}_2^{S2}$	$\mathbf{c}_2^{S2}$
$\mathbf{C_z}$	$H(h_a)C$	$H(h_a)C$	$H(h_a)C$	$H(h_a)C$
$\mathcal{H}^{\mathbf{opt}}$	$[h_4, h_{13}, h_{16}, h_{17}, h_{19}]$	$[h_3, h_{13}, h_{15}, h_{16}, h_{17}, h_{19}]$	$[h_3, h_{12}, h_{15}, h_{17}, h_{18}, h_{20}]$	$[h_3, h_{12}, h_{14}, h_{15}, h_{16}, h_{17}, h_{18}]$
$\lvert\mathcal{H}^{\mathbf{opt}}\rvert$	5	6	6	7
α^{opt}	0.0671	0.647	0.0031	0.0026
τ	0.30[sec]	0.28[sec]	0.34[sec]	0.3531[sec]
$\lVert\mathbf{e}\rVert_\infty^{\mathbf{L_1}}$	0.0967	0.0966	0.1124	0.1123
$\lVert\mathbf{e}\rVert_\infty^{\mathbf{H_2}}$	1.3720	1.5649	1.7796	1.8638

Table 5.2: Simulative results obtained trough the proposed approach (solution to the optimization problem (5.18)-(5.23)): objective function (5.18) weighting parameters ($\mathbf{a_1}$, $\mathbf{a_2}$, $\mathbf{c}$), objective vector matrix (C_z), optimal sensor set ($\mathcal{H}^{opt}$), optimal sensor set cardinality ($|\mathcal{H}^{opt}|$), bound of the optimal L_1 norm (α^{opt}), execution time (τ), ∞-norm of the state reconstruction error for the L_1 observer ($\|e\|_\infty^{L_1}$) and ∞-norm of the state reconstruction error for the H_2 observer ($\|e\|_\infty^{H_2}$).

the state reconstruction error for the L_1 and H_2 observers, it is possible to confirm that the L_1 positive observer proposed in this study shows better state reconstruction error performance than the standard H_2 observer for the same set $\mathcal{H}^{opt}$ achieved in every Test 1-8. Further analyses have been performed to contrast the proposed procedure with the following alternative approaches:

1. Simulated annealing sensor selection procedure introduced in Chapter 3;
2. Genetic algorithm sensor selection procedure proposed in [17];

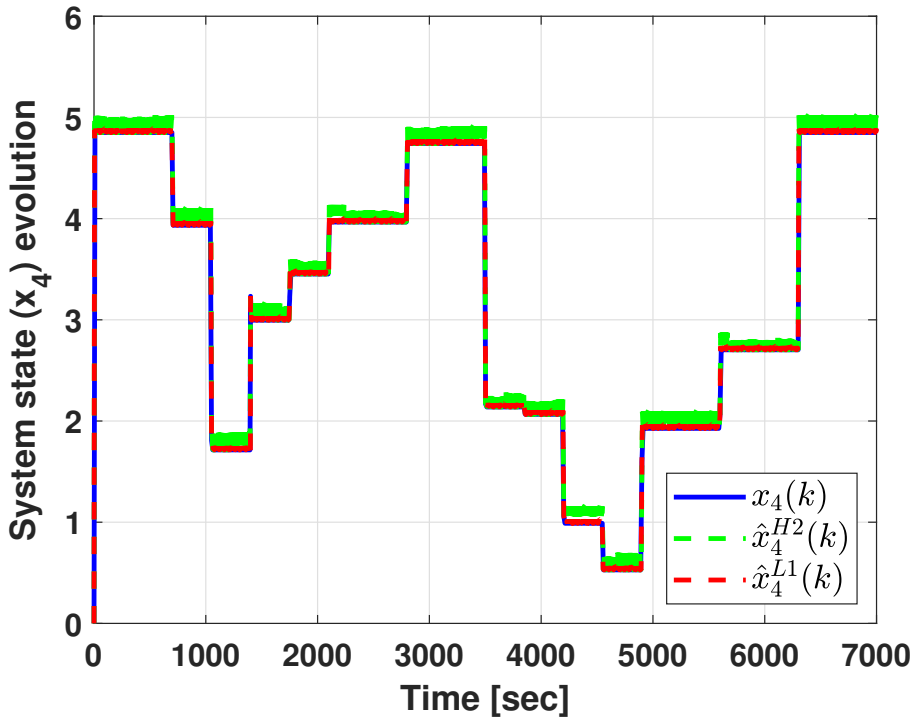

FIGURE 5.3: System state evolution: system state x_4 (blue line), system state estimate $\hat{x}_4^{L1}$ (red dashed line) provided by the proposed L_1 observer and system state estimate $\hat{x}_4^{H2}$ (green dashed line) provided by the standard H_2 observer.

Also for these further analyses the disturbance sequence ($w(k)$) was considered. Specifically, two Luenberger observers have been developed by considering the sensor sets obtained through the sensor selection process using SA and GA techniques and the results have been summarized in Table 5.3. This Table reports the computed optimal sensor set ($\mathcal{H}^{opt}$) and the optimal sensor set cardinality ($|\mathcal{H}^{opt}|$), the selection procedure execution time (τ) and the ∞-norm of the state reconstruction error ($\|e\|_\infty^{H_2}$) for both sensor selections determined by the SA and GA procedures. The results suggest that the proposed L_1-based approach outperforms a standard H_2 Luenberger observer that considers the sensor sets obtained through a heuristic procedure.

From the obtained results it is evident that the proposed strategy:

- remarkably reduces the execution times w.r.t. SA and GA algorithms;
- allows the achievement of better solutions in terms of both sensor set cardinality and state reconstruction error compared to the sensor selection procedure based on a combination of mixed-integer programming and H_2 observer design (see the results reported in Tables 5.2 and 5.3). Furthermore, from Tables 5.2 and 5.3, it is evident that the approach based on H_2 principles is comparable in terms of sensor set cardinality to the proposed

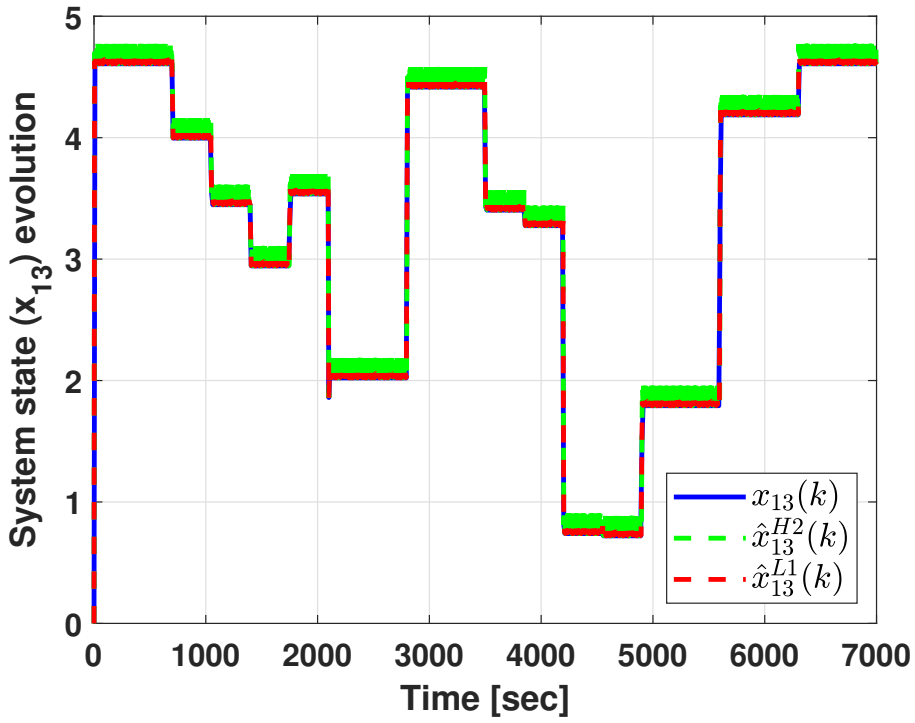

FIGURE 5.4: System state evolution: system state x_{13} (blue line), system state estimate $\hat{x}_{13}^{L1}$ (red dashed line) provided by the proposed L_1 observer and system state estimate $\hat{x}_{13}^{H2}$ (green dashed line) provided by the standard H_2 observer.

Test	$\mathcal{H}^{\mathbf{opt}}$	$\lvert\mathcal{H}^{\mathbf{opt}}\rvert$	$\mathcal{K}[W(h)]$	τ	$\lVert\mathbf{e}\rVert_{\infty}^{\mathbf{H_2}}$
SA	$[h_4, h_{12}, h_{15}, h_{17}, h_{19}]$	5	4.1002×10^3	95.6[sec]	3.623
GA	$[h_1, h_4, h_{12}, h_{15}, h_{16}, h_{19}]$	6	7.584×10^4	89.5[sec]	3.255

Table 5.3: Simulative results obtained trough the alternative approaches (simulated annealing (SA) and genetic algorithm (GA) procedures): optimal sensor set ($\mathcal{H}^{opt}$), optimal sensor set cardinality ($|\mathcal{H}^{opt}|$), execution time (τ) and ∞-norm of the state reconstruction error ($\|e\|_{\infty}^{H_2}$).

approach only in Test 5. However, this solution requires more execution time.

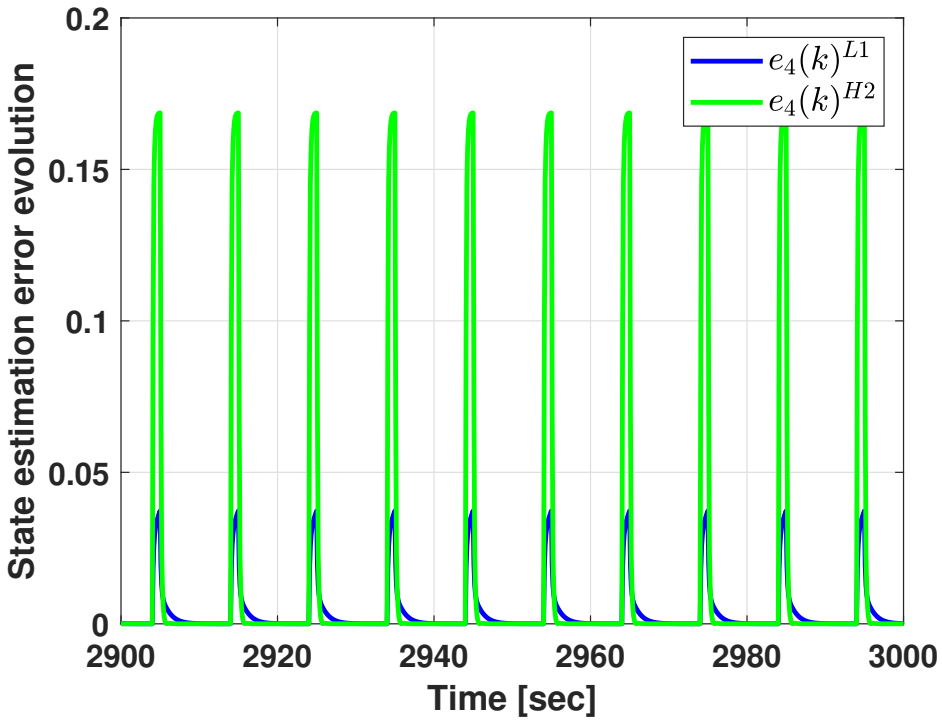

FIGURE 5.5: State reconstruction error evolution for $k \in [2900, 3000]$ [sec]: $e_4^{L1}(k)$ (blue line) and $e_4^{H2}(k)$ (green line).

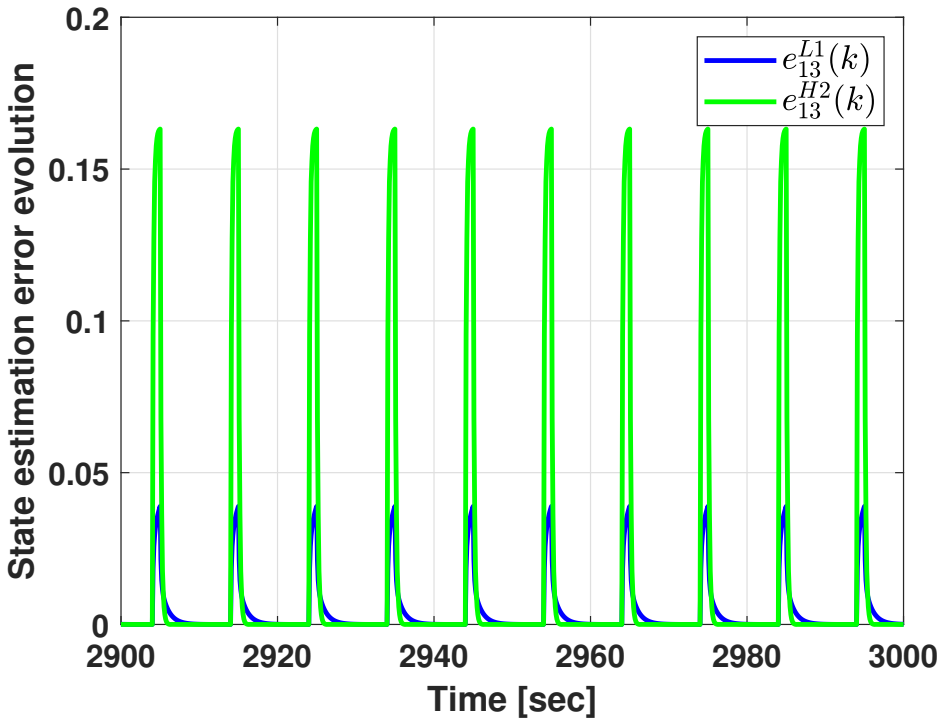

FIGURE 5.6: State reconstruction error evolution for $k \in [2900, 3000]$ [sec]: $e_{13}^{L1}(k)$ (blue line) and $e_{13}^{H2}(k)$ (green line).

5.4 Conclusions

In this Chapter, the joint sensor selection and L_1 positive observer design problem was formulated for positive systems via mixed-integer semi-definite optimization leveraging system Observability. In order to reduce the error in state reconstruction, a crucial performance metric for positive observers was the use of L_1 optimality. The theoretical background of positive L_1 optimum observer design techniques for positive systems was first examined. Essential criteria to ensure the asymptotic stability of the observer, along with establishing a bound on the state reconstruction error were established. To validate the efficacy of the proposed method, simulations have been conducted and comparisons with the results achieved by a classical H_2 observer and genetic and simulated annealing algorithms have been also reported.

Finally, future efforts will focus on applying the proposed procedure to traffic flows monitoring for smart lighting system applications. A smart lighting system is a technological framework designed to automatically adjust the brightness of street lamps based on real-time vehicular movement in designated areas, thereby enhancing energy efficiency. Moreover, it is noteworthy that our approach, in the context of traffic management, places emphasis on the creation of a traffic flow estimation model that is efficient and requires fewer monitoring equipment. By lowering the number of sensors required, this method provides an economical yet effective way to enhance system efficiency while maintaining accurate traffic situation assessments.

Conclusions and Future Work 6

This dissertation has investigated the fault tolerant estimation problem through a two-dimensional perspective: (i) design of reliable estimation schemes based on the Sensor Reconciliation methodology suitable for both centralized and distributed scenarios; (ii) investigation and implementation of procedures for the selection of the optimal number/locations of sensors to efficiently execute a reliable estimation task. On this regard, the main contributions are here listed and directions for future research are provided as well.

6.1 Conclusions

The main contributions of this thesis with respect to the existing literature on fault-tolerant estimation schemes and sensor selection methodologies are as follows:

- **Centralized Sensor Reconciliation**: Chapter 3 introduced a centralized Sensor Reconciliation architecture that merges traditional Observer and Reconciliator Units into a single block. This unified approach leverages solely system input and output information to estimate internal state and faulty parameters. The estimated parameters are then utilized within a self-tuning technique to mitigate the impact of faults and ensure reliable output. The architecture further exploits physical or analytical redundancy to enhance reliability. To address the sensor section problem, a sensor selection methodology based on an Observability optimization problem and a heuristic solution was discussed. The architecture's effectiveness was demonstrated through extensive simulations and experimental scenarios, including its application to a road network modeled as a compartmental system for reliable traffic data estimation.

- **Distributed Sensor Reconciliation**: Chapter 4 presented a Distributed Sensor Reconciliation methodology, derived from a decomposed centralized steady-state Kalman Filter. This approach enables distributed state estimation of a Linear Time Invariant plant in the presence of unpredictable sensor faults. The scenario considered involved individual local measurements that may not individually guarantee system observability, but collectively did. To address this challenge, the proposed DSR scheme aimed

at masking faulty sensor measurements within local state estimates, ensuring convergence to the true system state. This was accomplished by fusing local measurements with communicated estimates from a subset of neighboring agents at each site. By leveraging sensor redundancy, the estimation scheme enhanced robustness against both multiplicative and additive sensor faults. Theoretical stability guarantees were provided, and simulation examples, including its application to a road network modeled as a compartmental system for reliable traffic data estimation, were discussed to illustrate the scheme's effectiveness.

- **Sensor Selection**: Chapter 5 presented a unified approach for sensor selection and observer design for positive systems. By leveraging positive system properties and algebraic relaxing methods, a convex Mixed Integer Quadratic Optimization problem was formulated. Solving this problem yields both the optimal set of sensors for estimation and the subsequent optimal L_1 observer. The problem was structured to prioritize either minimizing the number of sensors or reducing the estimation error of the L_1 observer. The feasibility of the optimization problem was analyzed, and the optimality of the solutions was proven. In conclusion, the methodology's effectiveness was extensively demonstrated through simulations.

6.2 Future Research

Although the results underling this research thesis has achieved the main goal of proposing a centralized and a distributed sensor reconciliation scheme and a sensor selection methodology some issues remain unsolved and a lot of room for future applications and extensions do exist:

- A real-world application of the centralized sensor reconciliation scheme to on-field traffic data estimation would serve as a crucial validation step towards realistic applications and hopefully industrial acceptance. In this context, efforts are currently being made to integrate such a scheme into a smart lighting architecture.
- While external distributed units can be employed to estimate multiplicative faults, a significant advancement for the distributed sensor reconciliation architecture would involve the ability to autonomously estimate these parameters. Ongoing efforts are focused on achieving this objective;
- Future efforts regarding the joint sensor selection and observer design methodology will focus on applying the proposed procedure to a realistic scenario of traffic flow monitoring for smart lighting system applications.

6.3 Research Acknowledgment

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART").

Bibliography

[1] K. Djath, M. Dufaut, and D. Wolf, "Mobile robot multisensor reconfiguration," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No. 00TH8511)*, IEEE, Dearborn, MI, USA, 2000, pp. 110–115.

[2] A. Mirabadi, F. Schmid, and N. Mort, "Multisensor integration methods in the development of a fault-tolerant train navigation system," *The Journal of Navigation*, vol. 56, no. 3, pp. 385–398, 2003.

[3] I. Bessa, V. Puig, and R. M. Palhares, "Reconfiguration blocks and fault hiding: Design, applications, and challenges," *Annual Reviews in Control*, vol. 56, p. 100 896, 2023.

[4] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual reviews in control*, vol. 32, no. 2, pp. 229–252, 2008.

[5] J. A. De Doná, M. M. Seron, and A. Yetendje, "Multisensor fusion fault-tolerant control with diagnosis via a set separation principle," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, IEEE, Shangai, China, 2009, pp. 7825–7830.

[6] A. Yetendje, J. A. De Doná, and M. M. Seron, "Multisensor fusion fault tolerant control," *Automatica*, vol. 47, no. 7, pp. 1461–1466, 2011.

[7] C. Berbra, S. Lesecq, and J. Martinez, "A multi-observer switching strategy for fault-tolerant control of a quadrotor helicopter," in *2008 16th Mediterranean Conference on Control and Automation*, IEEE, Ajaccio, France, 2008, pp. 1094–1099.

[8] M. Romero, M. Seron, and J. De Dona, "Sensor fault-tolerant vector control of induction motors," *IET control theory & applications*, vol. 4, no. 9, pp. 1707–1724, 2010.

[9] H. Behzad, A. Casavola, F. Tedesco, and M. A. Sadrnia, "A fault-tolerant sensor reconciliation scheme based on lpv unknown input observers," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, Las Vegas, NV, USA, 2016, pp. 2158–2163.

[10] H. Behzad, A. Casavola, F. Tedesco, and M. A. Sadrnia, "Fault-tolerant sensor reconciliation schemes based on unknown input observers," *International Journal of Control*, vol. 93, no. 3, pp. 669–679, 2020.

[11] E. Chow and A. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Transactions on automatic control*, vol. 29, no. 7, pp. 603–614, 1984.

[12] X. Ding, L. Guo, and T. Jeinsch, "A characterization of parity space and its application to robust fault detection," *IEEE Transactions on Automatic Control*, vol. 44, no. 2, pp. 337–343, 1999.

[13] J. Gertler and D. Singer, "A new structural framework for parity equation-based failure detection and isolation," *Automatica*, vol. 26, no. 2, pp. 381–388, 1990.

[14] K. Manohar, J. N. Kutz, and S. L. Brunton, "Optimal sensor and actuator selection using balanced model reduction," *IEEE Transactions on Automatic Control*, vol. 67, no. 4, pp. 2108–2115, 2021.

[15] M. Fabris, R. Ceccato, and A. Zanella, "Efficient sensors selection for traffic flow monitoring: An overview of model-based techniques leveraging network observability," *arXiv preprint arXiv:2404.08588*, 2024.

[16] G. Baggio, F. Pasqualetti, and S. Zampieri, "Energy-aware controllability of complex networks," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 465–489, 2022.

[17] L. Varotto, A. Zampieri, and A. Cenedese, "Street sensors set selection through road network modeling and observability measures," in *2019 27th Mediterranean Conference on Control and Automation (MED)*, IEEE, Akko, Israel, 2019, pp. 392–397.

[18] T. H. Summers and J. Lygeros, "Optimal sensor and actuator placement in complex dynamical networks," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3784–3789, 2014.

[19] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 91–101, 2015.

[20] R. Li, N. Mehr, and R. Horowitz, "Submodularity of optimal sensor placement for traffic networks," *Transportation research part B: methodological*, vol. 171, pp. 29–43, 2023.

[21] G. Gagliardi, A. Casavola, and V. D'Angelo, "Traffic sensors selection for complete link flow observability through simulated annealing," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 10 540–10 545, 2023.

[22] M. Staroswiecki, G. Hoblos, and A. Aitouche, "Sensor network design for fault tolerant estimation," *International journal of adaptive control and signal processing*, vol. 18, no. 1, pp. 55–72, 2004.

[23] E. Clark, J. N. Kutz, and S. L. Brunton, "Sensor selection with cost constraints for dynamically relevant bases," *IEEE Sensors Journal*, vol. 20, no. 19, pp. 11 674–11 687, 2020.

[24] K. K. Chen and C. W. Rowley, "H2 optimal actuator and sensor placement in the linearised complex ginzburg–landau system," *Journal of Fluid Mechanics*, vol. 681, pp. 241–260, 2011.

[25] T. Summers and I. Shames, "Convex relaxations and gramian rank constraints for sensor and actuator selection in networks," in *2016 IEEE International Symposium on Intelligent Control (ISIC)*, IEEE, Buenos Aires, Argentina, 2016, pp. 1–6.

[26] K. Yamada, Y. Sasaki, T. Nagata, *et al.*, "Efficient sensor node selection for observability gramian optimization," *Sensors*, vol. 23, no. 13, p. 5961, 2023.

[27] M. Siami, A. Olshevsky, and A. Jadbabaie, "Deterministic and randomized actuator scheduling with guaranteed performance bounds," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1686–1701, 2020.

[28] S. D. Bopardikar, "Sensor selection via randomized sampling," *arXiv preprint arXiv:1712.06511*, 2017.

[29] N. K. Dhingra, M. R. Jovanović, and Z.-Q. Luo, "An admm algorithm for optimal sensor and actuator selection," in *53rd IEEE Conference on Decision and Control*, IEEE, Los Angeles, CA, USA, 2014, pp. 4039–4044.

[30] V. Tzoumas, L. Carlone, G. J. Pappas, and A. Jadbabaie, "Lqg control and sensing co-design," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1468–1483, 2020.

[31] A. Zare, H. Mohammadi, N. K. Dhingra, T. T. Georgiou, and M. R. Jovanović, "Proximal algorithms for large-scale statistical modeling and sensor/actuator selection," *IEEE Transactions on Automatic Control*, vol. 65, no. 8, pp. 3441–3456, 2019.

[32] J. Milošević, A. Teixeira, K. H. Johansson, and H. Sandberg, "Actuator security indices based on perfect undetectability: Computation, robustness, and sensor placement," *IEEE Transactions on Automatic Control*, vol. 65, no. 9, pp. 3816–3831, 2020.

[33] M. Pirani, E. Nekouei, H. Sandberg, and K. H. Johansson, "A game-theoretic framework for security-aware sensor placement problem in networked control systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3699–3706, 2021.

[34] J. Qin, Q. Ma, Y. Shi, and L. Wang, "Recent advances in consensus of multi-agent systems: A brief survey," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4972–4983, 2016.

[35] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[36] Y. Wang, J. Cao, and A. Kashkynbayev, "Multi-agent bifurcation consensus based multi-layer uavs formation keeping control and its visual simulation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 8, pp. 3221–3233, 2023.

[37] W. M. Haddad, V. Chellaboina, and Q. Hui, *Nonnegative and compartmental dynamical systems*. Princeton University Press, 2010.

[38] J. D. Murray, *Mathematical biology: I. An introduction*. Springer Science & Business Media, 2007, vol. 17.

[39] R. Shorten, F. Wirth, and D. Leith, "A positive systems model of tcp-like congestion control: Asymptotic results," *IEEE/ACM transactions on networking*, vol. 14, no. 3, pp. 616–629, 2006.

[40] J. Van Den Hof, "Positive linear observers for linear compartmental systems," *SIAM Journal on Control and Optimization*, vol. 36, no. 2, pp. 590–608, 1998.

[41] A. McGlinchey and O. Mason, "Some novel aspects of the positive linear observer problem: Differential privacy and optimal l1 sensitivity," *Journal of the Franklin Institute*, vol. 357, no. 18, pp. 13 923–13 940, 2020.

[42] C. Briat and M. Khammash, "Interval peak-to-peak observers for continuous and discrete-time systems with persistent inputs and delays," *Automatica*, vol. 74, pp. 206–213, 2016.

[43] M. A. Rami, F. Tadeo, and U. Helmke, "Positive observers for linear positive systems, and their implications," *International Journal of Control*, vol. 84, no. 4, pp. 716–725, 2011.

[44] T. Steffen, *Control reconfiguration of dynamical systems: linear approaches and structural tests*. Springer Science & Business Media, 2005, vol. 320.

[45] D. Rotondo, F. Nejjari, and V. Puig, "A virtual actuator and sensor approach for fault tolerant control of lpv systems," *Journal of Process Control*, vol. 24, no. 3, pp. 203–222, 2014.

[46] D. Rotondo, F. Nejjari, and V. Puig, "Fault tolerant control of a proton exchange membrane fuel cell using takagi–sugeno virtual actuators," *Journal of Process Control*, vol. 45, pp. 12–29, 2016.

[47] A. S. Pedersen, J. H. Richter, M. Tabatabaeipour, H. Jóhannsson, and M. Blanke, "Fault tolerant emergency control to preserve power system stability," *Control Engineering Practice*, vol. 53, pp. 151–159, 2016.

[48] D. Rotondo, V. Puig, F. Nejjari, and J. Romera, "A fault-hiding approach for the switching quasi-lpv fault-tolerant control of a four-wheeled omnidirectional mobile robot," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3932–3944, 2014.

[49] J. H. Richter, *Reconfigurable control of nonlinear dynamical systems: a fault-hiding approach*. Springer, 2011, vol. 408.

[50] I. Bessa, V. Puig, and R. M. Palhares, "Passivation blocks for fault tolerant control of nonlinear systems," *Automatica*, vol. 125, p. 109 450, 2021.

[51] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[52] R. C. Luo, Y. C. Chou, and O. Chen, "Multisensor fusion and integration: Algorithms, applications, and future research directions," in *2007 International Conference on Mechatronics and Automation*, IEEE, Harbin, China, 2007, pp. 1986–1991.

[53] P. J. Nahin and J. L. Pokoski, "Nctr plus sensor fusion equals iffn or can two plus two equal five?" *IEEE transactions on Aerospace and Electronic systems*, no. 3, pp. 320–337, 1980.

[54] R. Nasburg and K. Moravec, "Distributed multisensor data fusion," in *Aircraft Design Systems and Operations Meeting*, San Diego, CA, USA, 1984, p. 2477.

[55] R. C. Luo, M.-H. Lin, and R. S. Scherp, "Dynamic multi-sensor data fusion system for intelligent robots," *IEEE Journal on Robotics and Automation*, vol. 4, no. 4, pp. 386–396, 1988.

[56] B. C. Moxon, "A multiprocessor-based sensor fusion software architecture," in *Automated Inspection and High-Speed Vision Architectures*, SPIE, vol. 849, Cambridge, CA, United States, 1988, pp. 72–79.

[57] E. L. Waltz and D. M. Buede, "Data fusion and decision support for command and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 6, pp. 865–879, 1986.

[58] A. Mitra and S. Sundaram, "Distributed observers for lti systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3689–3704, 2018.

[59] R. Olfati-Saber, "Distributed kalman filtering for sensor networks," in *2007 46th IEEE Conference on Decision and Control*, IEEE, New Orleans, LA, USA, 2007, pp. 5492–5498.

[60] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, Ieee, Shanghai, China, 2009, pp. 7036–7042.

[61] B. Shen, Z. Wang, and Y. S. Hung, "Distributed h∞-consensus filtering in sensor networks with multiple missing measurements: The finite-horizon case," *Automatica*, vol. 46, no. 10, pp. 1682–1688, 2010.

[62] V. Ugrinovskii, "Distributed robust estimation over randomly switching networks using h∞ consensus," *Automatica*, vol. 49, no. 1, pp. 160–168, 2013.

[63] J. Yan, X. Yang, Y. Mo, and K. You, "A distributed implementation of steady-state kalman filter," *IEEE Transactions on Automatic Control*, vol. 68, no. 4, pp. 2490–2497, 2022.

[64] M. Farina, G. Ferrari-Trecate, and R. Scattolini, "Distributed moving horizon estimation for linear constrained systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2462–2475, 2010.

[65] L. Wang and A. S. Morse, "A distributed observer for a time-invariant linear system," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 2123–2130, 2017.

[66] K. Schenk and J. Lunze, "Fault-tolerant control in networked systems: A two-layer approach," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, Melbourne, VIC, Australia, 2017, pp. 6370–6376.

[67] G. Yang, H. Rezaee, A. Serrani, and T. Parisini, "Sensor fault-tolerant state estimation by networks of distributed observers," *IEEE Transactions on Automatic Control*, vol. 67, no. 10, pp. 5348–5360, 2022.

[68] D. Rotondo, D. Theilliol, and J.-C. Ponsart, "Virtual actuator and sensor fault tolerant consensus for homogeneous linear multi-agent systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2023.

[69] G. Gagliardi, V. D'Angelo, F. A. Torchiaro, and A. Casavola, "Trustworthy traffic sensors networks via fault-tolerant sensor reconciliation architectures," *IEEE Transactions on Automation Science and Engineering*, pp. 1–15, 2024. DOI: 10.1109/TASE.2024.3395980.

[70] G. Gagliardi, F. A. Torchiaro, and A. Casavola, "Joint sensor selection and observer design for positive systems via mixed-integer semidefinite programming," *IEEE Control Systems Letters*, 2024.

[71] G. Gagliardi, F. A. Torchiaro, and A. Casavola, "Joint sensor selection and observer design for positive systems via mixed-integer semidefinite programming," in *63rd IEEE international Conference on Decision and Control (CDC)*, IEEE, Milan, Italy, 2024.

[72] F. A. Torchiaro, G. Gagliardi, F. Tedesco, A. Casavola, and B. Sinopoli, "A fault-tolerant distributed sensor reconciliation scheme based on decomposed steady-state kalman filter," in *63rd IEEE international Conference on Decision and Control (CDC)*, IEEE, Milan, Italy, 2024.

[73] D. Luenberger, "Observers for multivariable systems," *IEEE transactions on automatic control*, vol. 11, no. 2, pp. 190–197, 1966.

[74] R. Isermann and P. Balle, "Trends in the application of model-based fault detection and diagnosis of technical processes," *Control engineering practice*, vol. 5, no. 5, pp. 709–719, 1997.

[75] G. J. Ducard, *Fault-tolerant flight control and guidance systems: Practical methods for small unmanned aerial vehicles*. Springer Science & Business Media, 2009.

[76] E. Sobhani-Tehrani and K. Khorasani, *Fault diagnosis of nonlinear systems using a hybrid approach*. Springer, 2009, vol. 383.

[77] M. Zhang, "Fault diagnosis & root cause analysis of invertible dynamic system," Ph.D. dissertation, Université Paul Sabatier-Toulouse III, 2017.

[78] X. Han, "Observer based fault diagnosis and fault tolerant control of nonlinear systems," Ph.D. dissertation, INSA de Toulouse, 2021.

[79] J. S. Shamma, "Analysis and design of gain scheduled control systems," Ph.D. dissertation, Massachusetts Institute of Technology, 1988.

[80] J. S. Shamma and M. Athans, "Gain scheduling: Potential hazards and possible remedies," *IEEE Control Systems Magazine*, vol. 12, no. 3, pp. 101–107, 1992.

[81] E. Garone, D. Talia, and A. Casavola, "Model predictive control schemes for linear parameter varying systems," Ph.D. dissertation, University of Calabria, 2014.

[82] P. Maréchal and J. J. Ye, "Optimizing condition numbers," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 935–947, 2009.

[83] M. Witczak, V. Puig, D. Rotondo, and P. Witczak, "A necessary and sufficient condition for total observability of discrete-time linear time-varying systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 729–734, 2017.

[84] S. Hanba, "On the "uniform" observability of discrete-time nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1925–1928, 2009.

[85] R. J. Caverly and J. R. Forbes, "Lmi properties and applications in systems, stability, and control theory," *arXiv preprint arXiv:1903.08599*, 2019.

[86] P. Gahinet and P. Apkarian, "A linear matrix inequality approach to h∞ control," *International journal of robust and nonlinear control*, vol. 4, no. 4, pp. 421–448, 1994.

[87] B. T. Hinson, "Observability-based guidance and sensor placement," Ph.D. dissertation, University of Washington, 2014.

[88] A. Stetco, F. Dinmohammadi, X. Zhao, *et al.*, "Machine learning methods for wind turbine condition monitoring: A review," *Renewable energy*, vol. 133, pp. 620–635, 2019.

[89] A. Olshevsky, "Minimal controllability problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, 2014.

[90] M. Gendreau, J.-Y. Potvin, *et al.*, *Handbook of metaheuristics*. Springer, 2010, vol. 2.

[91] G. Gagliardi, A. Casavola, and V. D'Angelo, "Reliable traffic sensor networks via fault-tolerant sensor reconciliation schemes," *IFAC PapersOnLine*, vol. 56, no. 2, pp. 10 558–10 563, 2023.

[92] G. G. Walter and M. Contreras, *Compartmental modeling with networks*. Springer Science & Business Media, 1999.

[93] L. Ingber, "Adaptive simulated annealing (asa): Lessons learned," *arXiv preprint cs/0001018*, 2000.

[94] M. M. Quadros, V. J. Leite, and R. M. Palhares, "Robust fault hiding approach for t–s fuzzy systems with unmeasured premise variables," *Information Sciences*, vol. 589, pp. 690–715, 2022.

[95] S. Leonardi and N. Distefano, "Turbo-roundabouts as an instrument for improving the efficiency and safety in urban area: An italian case study," *Sustainability*, vol. 15, no. 4, p. 3223, 2023.

[96] F. Vrbanić, D. Čakija, K. Kušić, and E. Ivanjko, "Traffic flow simulators with connected and autonomous vehicles: A short review," *Transformation of Transportation*, pp. 15–30, 2021.

[97] A. Pashkevich, J. Dyrcz, and O. Dubiel, "Modelling of runway capacity for the international airport krakow-balice using aimsun next software," in *Advances in Air Traffic Engineering: Selected Papers from 6th International Scientific Conference on Air Traffic Engineering*, Springer, Warsaw, Poland, 2021, pp. 106–115.

[98] V. Gallelli, T. Iuele, R. Vaiana, and A. Vitale, "Investigating the transferability of calibrated microsimulation parameters for operational performance analysis in roundabouts," *Journal of Advanced Transportation*, vol. 2017, no. 1, p. 3 078 063, 2017.

[99] V. Gallelli, G. Guido, A. Vitale, and R. Vaiana, "Effects of calibration process on the simulation of rear-end conflicts at roundabouts," *Journal of traffic and transportation engineering (English edition)*, vol. 6, no. 2, pp. 175–184, 2019.

[100] F. Boem, R. M. Ferrari, and T. Parisini, "Distributed fault detection and isolation of continuous-time non-linear systems," *European Journal of Control*, vol. 17, no. 5-6, pp. 603–620, 2011.

[101] D. Rotondo, J.-C. Ponsart, D. Theilliol, F. Nejjari, and V. Puig, "A virtual actuator approach for the fault tolerant control of unstable linear systems subject to actuator saturation and fault isolation delay," *Annual reviews in Control*, vol. 39, pp. 68–80, 2015.

[102] W. L. Hallauer Jr, *Introduction to linear, time-invariant, dynamic systems for students of engineering*. Virginia Tech, 2016.

[103] B. Anderson and J. Moore, *Optimal Filtering* (Dover Books on Electrical Engineering). Dover Publications, 2012, ISBN: 9780486136899.

[104] C. K. Liew, "Inequality constrained least-squares estimation," *Journal of the American Statistical Association*, vol. 71, no. 355, pp. 746–751, 1976.

[105] D. Liberzon and A. Morse, "Basic problems in stability and design of switched systems," *IEEE Control Systems Magazine*, vol. 19, no. 5, pp. 59–70, 1999.

[106] F. İlhan and Ö. Karabacak, "Graph-based dwell time computation methods for discrete-time switched linear systems," *Asian Journal of Control*, vol. 18, no. 6, pp. 2018–2026, 2016.

[107] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: A survey of recent results," *IEEE Transactions on Automatic control*, vol. 54, no. 2, pp. 308–322, 2009.

[108] L. Farina and S. Rinaldi, *Positive linear systems: theory and applications*. John Wiley & Sons, 2011.

[109] W. M. Haddad and V. Chellaboina, "Stability and dissipativity theory for nonnegative dynamical systems: A unified analysis framework for biological and physiological systems," *Nonlinear Analysis: Real World Applications*, vol. 6, no. 1, pp. 35–65, 2005.

[110] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems," *Mathematical programming*, vol. 10, no. 1, pp. 147–175, 1976.

[111] S. A. Nugroho and A. F. Taha, "Sensor placement strategies for some classes of nonlinear dynamic systems via lyapunov theory," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, Nice, France, 2019, pp. 4551–4556.

[112] H. Wolkowicz, R. Saigal, and L. Vandenberghe, *Handbook of semidefinite programming: theory, algorithms, and applications*. Springer Science & Business Media, 2012, vol. 27.

[113] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, IEEE, Taipei, Taiwan, 2004, pp. 284–289.

[114] E. D. Andersen and K. D. Andersen, "The mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm," in *High performance optimization*, Boston, MA: Springer, 2000, pp. 197–232.

[115] A. Casavola, G. Franzè, G. Gagliardi, and F. Tedesco, "Improving lighting efficiency for traffic road networks: A reputation mechanism based approach," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 4, pp. 1743–1753, 2022.