Laboratório de Modelagem, Análise e Controle de Sistemas Não-Lineares

Departamento de Engenharia Eletrônica

Universidade Federal de Minas Gerais

Av. Antônio Carlos 6627, 31270-901 Belo Horizonte, MG Brasil

Fone: +55 3499-4866 - Fax: +55 3499-4850

Sensor Fusion for Irregularly Sampled Systems

Taiguara Melo Tupinambás

Dissertação submetida à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como parte dos requisitos necessários à obtenção do grau de Mestre em Engenharia Elétrica.

Orientadores: Prof. Bruno Otávio Soares Teixeira, Dr.
Prof. Leonardo Antônio Borges Tôrres, Dr.

Belo Horizonte, Fevereiro de 2019

Contents

Todo list

Resumo

Traduzir

Palavras-chave: Traduzir.

Abstract

The use of multiple sensors to improve data quality has grown continuously over the last few decades. With the never-ending advances in technology of microprocessors and communication devices, sensor networks will continue to increase in both size and complexity. The most popular applications for fusing data from various sources are related to estimating the states of a dynamic system. For that, two noisy sources of information are needed: a process model, that describes how the states evolve in time; and an observation model, whose data are usually derived from sensors. Since most sensors are digital, signals must be sampled in order to be processed, leading to sampled-data systems. Classical state estimators for sampled-data systems, like the well-known Kalman filter, consider the sampling process as the output of an ideal sampler. In other words, the source of information transmits data at regularly spaced time intervals. Under such consideration, continuous systems can be discretized into time-invariant representations in most cases. However, because of the widespread use of complex sensor networks without time synchronization, many applications cannot rely on data being transmitted regularly. There are adaptations to state estimation techniques that handle most of the irregularities, providing that timestamps are part of the measurement packet and that the increase in computational costs are acceptable. If timestamps cannot be used in the estimation process, one can either invest in synchronization or assimilate the information at incorrect time instants. The effects in estimation performance of the latter approach has not yet been extensively studied. Thus our work investigates how performance is deteriorated by neglecting measurements timestamps in state estimation algorithms. We consider the Poisson process as a model for the generation of the irregular time instants sequence, and study the errors introduced by shifting the signal samples in time. Then we simulate state estimation results for a linear and a nonlinear system with aperiodic sampling, using Kalman filter for the former and its unscented variation for the latter. Algorithms are designed to use timestamps in the estimation process and to neglect them, and their results over multiple runs are compared for different simulation scenarios. We observe situations where performance of both algorithms differ substantially and others where there are no statistically significant difference.

Keywords: Sensor Fusion; Irregular Sampling; State Estimation; Sampled-data Systems; Time Synchronization; Time-Stamp.

List of Figures

xiv

List of Tables

List of Symbols and Acronyms

Symbols

Chapter 1

$\mathbb{N}^+$ positive integers;

$\mathbb{R}$ real numbers;

$\mathbb{R}^n$ $\mathbb{R}^{nx1}$ n-dimensional Euclidean space;

$\forall$ for all;

$\in$ belongs to;

$>$ greater than;

$<$ less than;

$\geq$ greater than or equal to;

$\leq$ less than or equal to;

$\triangleq$ equals by definition;

$\approx$ approximately equal to;

$\sim$ is distributed as;

$\mathcal{E}(\lambda)$ exponential distribution with rate parameter λ;

$E[\cdot]$ mathematical expectation;

x_i ith entry of x;

f process model;

g observation model;

t continuous-time index;

k discrete-time index of measurements;

i discrete-time index of input;

t_k continuous-time sampled instants;

T	input sampling interval;
$x(t)$	state vector;
$\dot{x}(t)$	derivative of the state vector with respect to t;
$x(t_k)$	state vector at continuous-time sampled instants;
$u(t)$	input vector;
$u(iT)$	input vector at continuous-time sampled at time $t = iT$;
$w(t)$	process noise vector;
$y(t_k)$	output vector at continuous-time sampled instants;
δ_k	measurements time-delay;
$v(t_k)$	measurement noise vector;
h_k	time intervals between two continuous-time sampled instants;
λ	parameter of the exponential distribution;
λ_h	parameter of the exponentially distributed random variable h_k;
λ_{δ_k}	parameter of the exponentially distributed random variable δ_k;
N	amount of identical sensors in the sampling model;
L	sampling period of the identical sensors in the sampling model;
P	covariance matrix;
α	relation between output (expected) and input sampling time intervals .

Chapter 2

$\mathcal{N}(\mu,\sigma)$	Gaussian distribution with mean μ and standard deviation σ;
$\mathbb{P}$	power set;
μ	mean of a random variable;
σ	standard deviation of a random variable;
n	number of homogeneous sensors;
E	evidence;
H	hypothesis;
R_i	i^{th} fuzzy rule;
x_k	k^{th} input fuzzy variable;
$A_{i,j}$	j^{th} antecedent fuzzy set for the ith rule;
y_i	i^{th} consequent output variable for the ith rule;
C_i	consequent class for the ith rule;
X	input vector;

A	fuzzy set;
U	universe of discourse;
B	subset of features;
$B_*(X)$	*B-lower* approximation of *X*;
$B^*(X)$	*B-upper* approximation of *X*;
$BN_B(X)$	*B-boundary* region of *X*;
$\rho(x)$	probability density function of *x*;
$\rho(H|E)$	conditional probability density function of *H* given *E*;
$\pi_x(u)$	possibility distribution of *x* associated with *u*;
$\mu_A(x)$	membership function of *x* of a fuzzy se *A*;
⊆	is a subset of;
∩	set intersection;
∅	empty set;

Chapter 3

$\mathbb{N}$	natural numbers;
$\mathbb{R}$	real numbers;
$\mathbb{R}^n$	$\mathbb{R}^{nx1}$ *n*-dimensional Euclidean space;
$Ber(p)$	Bernoulli distribution with parameter *p*;
$\mathcal{N}(\mu,\sigma)$	Gaussian distribution with mean μ and standard deviation σ;
$\Gamma(\kappa,\theta)$	gamma distribution with shape parameter κ and scale parameter θ;
$E[\cdot]$	mathematical expectation;
$Var[\cdot]$	variance;
∀	for all;
>	greater than;
≥	greater than or equal to;
~	is distributed as;
±	plus or minus;
y_t, yk, ykT	output vector at continuous-time sampled instants;
$z(t)$	measurement output;
$H(k), L(k), C, D$	matrices of the observation model;
$v(t), v(k)$	measurement noise vector;
t_i, t_k	continuous-time sampled index;

k	discrete-time index of measurements;
T	expected sampling time interval;
$d(k)$	observation sampling time delays;
l	amount of different known delays;
$\xi(t)$	Bernoulli random variable that models packet dropout and uncertain observati
$C_i(t)$	real time approximation from the ith computer clock;
a_i	i^{th} clock drift;
b_i	i^{th} clock offset;
a_{ij}	relative drift between i^{th} and j^{th} node;
b_{ij}	relative offset between i^{th} and j^{th} node;
Δ	measured value interval between observations;
δ_k	random time interval;
$p(k)$	Bernoulli distribution parameter;
ϵ_k	deviation from expected time instants;
σ	standard deviation;
κ	shape parameter;
θ	scale parameter;
μs	micro seconds;
$E.E$	energy efficiency;
Comp.	complexity;

Chapter 4

$\mathbb{R}^n$	$\mathbb{R}^{nx1}$ n-dimensional Euclidean space;
$\mathcal{N}(\mu,\sigma)$	Gaussian distribution with mean μ and standard deviation σ;
$E[\cdot]$	mathematical expectation;
$\triangleq$	equals by definition;
$\arg\max\limits_{x} f(x)$	argument x that maximizes function $f(x)$;
$\rho(x\vert y)$	conditional probability density function of X given Y;
x_k	state vector;
$\hat{x_k}$	estimated state vector;
y_k	observation vector;
σ	standard deviation;
Σ	covariance matrix;

Chapter 5

Acronyms

CDF   Cumulative Distribution Function;

CU   Covariance Union;

DAI   Data In;

DAO   Data Out;

DEI   Decision In;

DEO   Decision Out;

DMTS   Delay Measurement Time Synchronization;

DSET   Dempster-Shafer Evidence Theory;

EKF   Extended Kalman Filter;

FEI   Feature In;

FEO   Feature Out;

FIS   Fuzzy Inference System;

FISST   Finite-Set Statistics;

FTSP   Flooding Time Synchronization Protocol;

GPS   Global Positioning System;

HMM   Hidden Markov Model;

KF   Kalman Filter;

UKF   Unscented Kalman Filter;

LEETS   Lightweight and Energy Efficient Time Synchronization;

LTS   Lightweight Tree-based Synchronization;

LTI   Linear Time-Invariant;

LS   Least Squares;

MAP   Maximum A Posteriori;

ML   Maximum Likelihood;

MMSE   Minimum Mean Square Error;

MOP   Measures Of Performance;

NEES   Normalized Estimation Error Squared;

NIS   Normalized Innovation Squared;

NMR   Nuclear Magnetic Resonance;

NTP   Network Time Protocol;

NUS   Non-Uniform Sampling;

OOSM   Out-Of-Sequence Measurement;

PDF	Probability Density Function;
PDAF	Probabilistic Data Association Filter;
PF	Particle Filter;
PRBS	Pseud-Random Binary Sequence;
RBS	Reference Broadcast Synchronization;
RMSE	Root Mean Square Error;
RTPF	Real-Time Particle Filter;
RV	Random Variable;
SOD	Send-On-Delta;
SNR	Signal-to-Noise Ratio;
TDS	Time-Delay System;
TDP	Time-Diffusion Protocol;
TPSN	Timing-sync Protocol for Sensor Networks;
TSST	Time Synchronization based on Spanning Tree;
TTF	Track-to-Track Fusion;
UKF	Unscented Kalman Filter;
WSN	Wireless Sensor Networks;
ZOH	Zero-Order Holder;

Introduction

1.1 Motivation

In nature it is possible to observe data fusion in a variety of phenomena. Animals combine signals from different senses, such as sight, hearing, smell, taste and touch, to recognize the surroundings. Plants have analogous mechanisms, which are used to modulate water consumption, to change the color of their leaves or to bend their structures towards the light, for instance. Throughout history, the sensory systems in living beings have evolved to assimilate multiple information coming from numerous sources in a highly complex and efficient way, in order to have a better perception of the environment.

Nowadays information fusion is studied in many fields of science, as a way of exploiting data from multiple sources to achieve better outcomes in comparison to those obtained if any of the sources were used separately (Dasarathy, 2001). Other terms have been used to denote the synthesis of information in technical literature, for instance, data fusion, sensor fusion, combination of evidence and synthesis of observations (Goodman et al., 1997). To avoid confusion, the terminology used by (Elmenreich, 2002) will be adopted, whereby information fusion is understood as the usage of any available information on the system being monitored and sensor fusion is used in cases for which the sources of information are sensor signals.

Some research fields have been increasingly making use of sensor fusion techniques, such as robotics, biometrics and image processing. The main benefits expected are related to improved *data authenticity*, by increasing accuracy, reliability and confidence, while reducing ambiguity and interference; or *data availability*, with higher spatial and temporal coverages and an increase in the perceived state space dimensionality, that is creating information by combining multiple available data. Consequently much effort has been devoted to developing and investigating data fusion techniques. The work of (Khaleghi et al., 2013) presents an extensive review of different available approaches,

categorizing them by the way sensor data imperfection is represented, namely, probabilistic fusion, evidential belief reasoning, fuzzy reasoning, possibilistic fusion, rough set-based fusion, random set-based fusion and hybrid fusion.

Data fusion techniques based on probability theory are the earliest available and perhaps the most popular until now. They are concerned with estimating the probability density functions (PDFs) of the system states by means of the Bayesian approach. If the system is linear and Gaussian, then the Kalman filter (KF) guarantees optimal estimation. For nonlinear processes, KF generalizations were proposed, such as the extended Kalman filter (EKF) or the unscented Kalman filter (UKF) (Julier and Uhlmann, 2004). On the other hand, particle filters (PF) can be used to deal with both nonlinearities in the dynamics and non-Gaussian distributions (Arulampalam et al., 2002).

The most common class of systems studied in state estimation is the class of sampled-data systems, due to the wide use of digital devices. Although often described by continuous time differential equations, they can be modeled using discrete-time state equations, based on approximation techniques (Phillips and Nagle, 1995). Usually the sampling period of such systems are constant and known. In other words, the sensors are considered to transmit data at regular intervals. However, for many applications, such assumption is not valid. The use of several redundant sensors, for example, with different sampling rates or not synchronized with one another, leads to data being received at irregular instants. Additionally, when data from multiple sensors are transmitted through several subsystems in a network, there might be loss of packets and delays (Schenato et al., 2007) or even multiple information arriving simultaneously (Moayedi et al., 2011). In networked control systems, event-triggered sampling schemes have been proposed to optimize the access to communication channels (Hu et al., 2017), which will also generate time-varying sampling intervals. Nowadays, because of the ever-growing scientific advances, the technology of microprocessors, sensors and communication has become increasingly accessible, which continues to ensure that multiple sensor networks are more and more common.

Thus, despite improving accuracy and robustness of the estimation process, the fusion of data from multiple sensors might introduce challenges to the state estimation algorithms, due to sampling irregularities. Depending on how they take place, modifications to the KF and its generalizations can be carried out to tackle these abnormalities. In the work of (Fatehi and Huang, 2017), the outputs of two individual KFs are fused to estimate the states of a system with multi-rate measurements, whereby one of them is

fast, regular and delay-free and the other is slow, irregular and randomly delayed. One application scenario is for industrial process control, where there is online instrumentation characterized by regularly sampled process signals together with asynchronous although very accurate data from laboratory analysis. For a more general case, when the random delays are unknown, the work of (Gopalakrishnan et al., 2010) presents a critical analysis of the available methods for data fusion. They are separated into two categories: those that incorporate the delayed measurements upon arrival, and methods that rely on state augmentation, in order to assimilate the delayed information between estimation steps.

In general the proposed methods and their performance will depend on the characteristics of the sampling irregularities and how they are modeled. Time delays can be multiples of a base sampling period, for instance. In those cases, delays can happen at single or multiple lags (Peñarrocha et al., 2012), they can lead to out-of-sequence measurements (Anxi et al., 2005; Westenberger et al., 2013) or there can also be data dropouts (Zhu et al., 2013). Nevertheless, the system can be described by a time-invariant discrete-time state equation, with particular representations of the observation model. When measurement are taken after random time intervals, the discrete-time state space representation leads to a time-varying system, since the sampling period changes over time. Some researchers treat the irregular measurement instants as stochastic processes (Micheli and Jordan, 2002) or as a periodic sampling interval subject to noisy perturbations (Shen et al., 2016). Generally, the time instant is considered to be the result of a measurement process and the methods assimilate such information in the algorithm, that is the random time intervals are considered as measurements themselves.

To the best of the author's knowledge, no method was proposed so far, for cases in which the random time instants or their statistics are not known or not reliable. If the sampling irregularity is caused due to the lack of sensor synchronization in the network, several algorithms can ensure a common timescale (Sivrikaya and Yener, 2004), at the expense of additional investments or energy use. Another approach, believed to be largely used in practice, is to simply disregard the irregularities, assimilating the measurements as soon as possible (Kwok et al., 2004; Huck et al., 2011). In this case, additional noise will appear in the measurement model as an outcome of not considering the correct time instants. In many cases the statistics of this additional noise is unknown and cannot be accounted for in the estimation process. However,

depending on system dynamics and parameters, it might be irrelevant to the overall performance.

Knowing to what extent the estimation accuracy is deteriorated by ignoring the additional uncertainty caused by the sampling irregularity is important to decide whether or not to invest in synchronization. In addition, the investment in more sensors sharing the same network in order to improve accuracy might not pay off, if it increases the occurrence of irregularities. However there are no detailed studies on the behavior of the degradation in accuracy due to neglecting irregularities in the sampling process. Therefore, this work assesses the differences in state estimation performance for systems sampled at random time intervals with and without time-stamp for different scenarios. The purpose is to shed some light on the trade-off for investments in sensor networks and synchronization.

1.2   Problem Formulation

Consider the stochastic nonlinear sampled system

$$\dot{x}(t) = f(x(t),u(t),w(t),t), \tag{1.1}$$

$$y(t_k) = g(x(t_k),v(t_k),t_k), \tag{1.2}$$

where $f\colon \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}^+ \to \mathbb{R}^n$ and $g\colon \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^+ \to \mathbb{R}^m$ are, respectively, the process and observation models, assumed to be known. We assume that for all $k \geq 1$, the observations $y(t_k) \in \mathbb{R}^m$ are available. Process and observation noises, $w(t)$ and $v(t_k)$ respectively, are white, Gaussian, zero-mean and mutually independent, with known covariance matrices. The first two moments of the initial random state vector $x(0)$ are also known. Input data $u(t)$ are available at regularly spaced time intervals T, that is $u(iT) \in \mathbb{R}^p$, $\forall i \geq 1$ are known.

Observations are taken at random time instants t_k and are considered to be sorted $(t_{k+1} > t_k, \forall k \in \mathbb{N}^+)$ and defined by the time intervals $h_0 \triangleq t_1$, $h_k \triangleq t_k - t_{k-1}$, $\forall k \geq 1$. In this work, we assume that the observation time instants t_k are given by a Poisson random process. In other words, the time intervals h_k are independent and identically distributed (i.i.d.) exponential random variables (RVs) with a known rate parameter λ, that is $h_k \sim \mathcal{E}(\lambda)$. Since the expected value of an exponential RV is the inverse of its rate

parameter, λ will sometimes be referred to as expected or average sampling rate of the irregular sampled quantity. An example of time intervals produced by such a random process is illustrated in Figure 1.1. This sampling model characterizes a common application for an event-based sampling scheme or for a networked control system with unsynchronized sensors. (Micheli and Jordan, 2002) consider a set of N identical sensors measuring the state variables of a physical process every L seconds. They prove that, if the sensors are independent and unsynchronized and N is large enough, then the waiting time between the realization of two consecutive measurements can be approximated by an exponential random variable $\mathcal{E}(\lambda)$, where the parameter is given by $\lambda = N/L$.
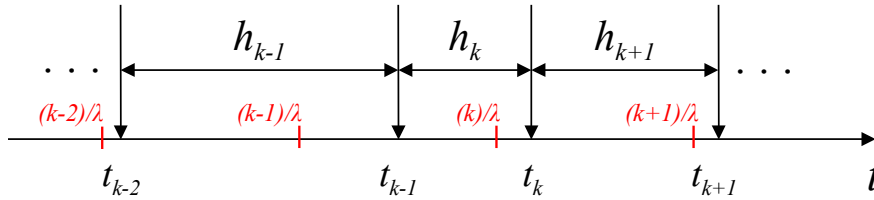
Figure 1.1: Irregular sampling process modeled by a Poisson random process. Regularly spaced time intervals $1/\lambda$ are shown in red. An example of time instants t_k realization is also shown, with the respective random time intervals h_k. The expected value of time interval is given by $E[h_k] = \frac{1}{\lambda}$.

When time-stamp information is available, data assimilation can be performed at the correct measurement instants t_k. When they are not, the assimilation moment is considered to be the random receipt time instant or the next estimation moment.

We wish to estimate the state vector $x(t)$ and its covariance recursively, at regularly spaced time intervals T, given their initial values x_0 and P_0, the process (1.1) and observation (1.2) models, the input or control signals, $u(iT) : iT \leq kT$, and the set of past observations, $y(t_k) : t_k \leq kT$. The knowledge of the time intervals $h_{k-1} : t_k \leq kT$ is also taken into consideration when time-stamp information is available. We assume that the average time interval of observations $1/\lambda$ is greater than or equal to T by a factor $\alpha \geq 1$, such that $1/\lambda = \alpha T$.

1.3   Objectives

Falta objetivos

1.4   Text Outline

This text is organized in six chapters, including this one. Chapter 1 presents an introduction to the object of study, with an overview of the motivations and historical perspective of the theme, objectives and problem formulation.

Chapter 2 presents a review of the sensor fusion field of science, addressing not only the definitions and taxonomy, but also the advantages of combining information from multiple sources. From the four categorization models of the data fusion problem, the one based on data challenges is further explored. We then introduce and discuss the methods proposed in literature to handle imperfect data.

Chapter 3 discusses the data-related challenges that arise for sampled-data systems, regarding sampling irregularities. Diagrams are built to organize the types, effects and causes of irregularities. We describe the necessary modifications to state estimation observation models, in order to handle such abnormalities in sampling schemes. To perform effective state estimation in the presence of sampling irregularities, methods depend on the knowledge of the exact time measurements were taken. Therefore, we explore time synchronization methods suited for sensor networks.

After the literature review, we focus on the probabilistic sensor fusion approach to sampled-data systems with sampling irregularities. The study of the impact of neglecting time-stamp information is performed.

Chapter 4 describes the methods used for simulation, considering the adaptations for the scenarios with and without time-stamp. We describe the filtering algorithm and the assumptions used for each scenario. Furthermore, we define the performance metrics used for the results assessment.

In Chapter 5 we present simulated results from two systems: an arbitrary linear system and a unicycle position estimation system. Signal parameters are varied to assess the impact in performance of both considering and not considering time-stamp in estimation algorithms. Performance is evaluated using estimated state errors and estimation consistency.

Finally, we conclude the study in Chapter 6, highlighting the study limitations and suggestions of future work, apart from evaluating if the proposed objectives were achieved.

Sensor Fusion

In this chapter, we review sensor fusion techniques, advantages and terminologies. We start with a brief explanation on the motivations of this field of science, grouping them in two main categories: data authenticity and data availability. We continue with a definition of sensor fusion and an exploration of the available fusion methods classification models. The categorization based on data challenges, that is uncertainty, imperfection, inconsistency and disparateness, is further discussed. We also present data fusion methods that handle imperfect data, which is the most fundamental problem present in information. We end the chapter focusing on probabilistic fusion methods for sampled-data systems.

2.1   Contextualization

The idea that combining information from multiple sensors to improve overall system performance has been in discussion for several decades. In the early days, there were those who argued against the synergism hype that was being spread in military systems, using the multi-sensor concept (Fowler, 1979). In his work, Fowler created what he called his seventh law:

> "Be wary of proposals for synergistic systems. Most of the time when you
> try to make 2 + 2 = 5, you end up with 3... and sometimes 1.9"

Although he was right to affirm that the added complexity and high costs are not always worth it - especially back then, when devices were more expensive - many posterior studies advocated that the fusion of sensor data will always be better, in the sense that the probability of correctly classifying a target increases. A direct answer to Fowler's work came in the year after (Nahin and Pokoski, 1980), when Nahin and Pokoski used strict concepts and definitions to prove that the addition of sensors improves network performance, but also acknowledged their assumptions of discarding

complexity and costs for the sake of their arguments. This topic continued to draw scientific attention throughout the years, like the work of (Rao, 1998) and (Dasarathy, 2000). Rao focused on fusion methods and its comparison to classifiers' performance. He also established conditions that guaranteed that the fused system will at least perform as good as the best classifier. Dasarathy's work extended that of Rao's, but it was able to describe a certain scenario at which a two-sensor scheme outperforms a three-sensor architecture from a parametric fusion benefits domain perspective. In order to compare performances of fusion levels or algorithms and assess possible benefits of sensor fusion, (Theil et al., 2000) discussed three measures of performance (MOP), one for each sensor management process, which are detection, tracking and classification.

Despite all philosophical discussions, many real applications have been taking advantage of sensor fusion benefits since its advent, like remote sensing (Foster and Hall, 1981), robotics (Richardson and Marsh, 1988) and intelligent systems (Luo and Kay, 1989). Recently, with the modernization and popularization of sensors, its use has grown significantly, with hot topics emerging in the area, such as body sensor networks for health-care applications (Gravina et al., 2017), artificial intelligence (Safari et al., 2014; Jordao et al., 2018) and smart grids (Liu and Wang, 2012; Kordestani and Saif, 2017). Recent reviews of the state of the art (Khaleghi et al., 2013; Jing et al., 2013) provide a very broad understanding of the field and its advances.

We continue this chapter advocating for the benefits of sensor fusion in Section 2.2, describing its motivations and advantages. In Section 2.3, most of the taxonomies used in the area are discussed. We end the chapter with a further discussion on the aspects of data fusion related to imperfect data.

2.2   Motivation and Advantages

Whether or not all sensor fusion approaches outperform the use of less sensors in every aspect for any given condition, the fact is that many fields of science and engineering have been benefiting from choosing the fusion approach.

The reasons why one chooses to fuse information from different sources are various. The works of (Hall and Llinas, 1997; Elmenreich, 2002; Andler and Brohede, 2009; Khaleghi et al., 2013) provide a detailed study on the motivations and advantages of multi-sensor data fusion techniques. A common benefit from the use of multiple

redundant sensors, for instance, is the increase in accuracy. By averaging all the measurements, the expected error decreases by the rate of $(\sqrt{n})^{-1}$, where n is the number of homogeneous sensors, in case of the presence of i.i.d Gaussian noises. In Figure 2.1 different probability density functions (PDFs) with different standard deviations - thus different expected errors - are shown for various quantities of random variables being averaged.
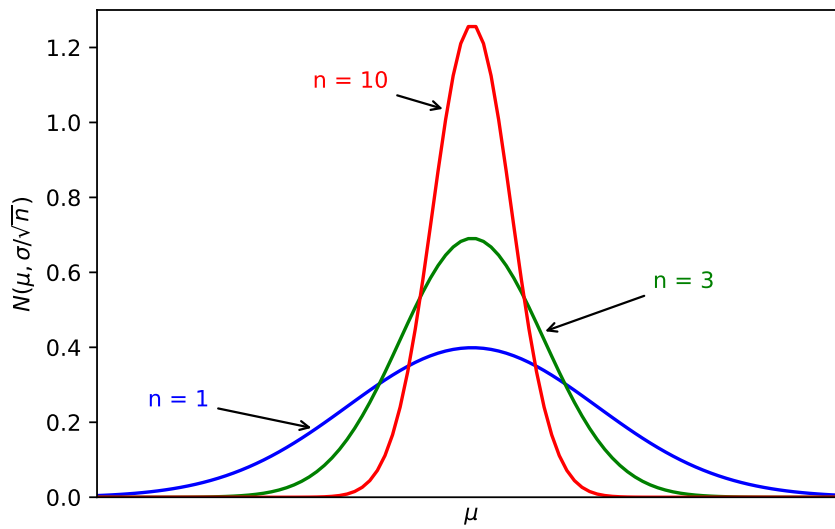
Figure 2.1: Probability distribution functions for different amount of fused measurements, obtained by 1 (blue), 3 (green) and 10 (red) redundant sensors, considering i.i.d additive Gaussian noise. The higher the value of n, the lower is the standard deviation, thus the lower is the expected measurement error.

There are many other reasons to fuse information from multiple sensors. (Elmenreich, 2002) provides an interesting example to illustrate some of them. Imagine a car parking assist system with only one distance sensor mounted at its rear, with limited precision and considerable update time (sampling interval). Now picture how would be the system's performance, considering that the sensor may suffer from: deprivation or occlusion, in the sense that it could be blocked by some physical barrier; coverage, both in spatial and temporal contexts, since it can only sense objects in front of him (spatial) and can only provide information periodically (temporal); precision, with measurement errors being liable to be the cause of unexpected bumps; and uncertainty, when an object, such as a small bicycle, might be missed by the sensor even in the

presence of valid signal. The system would definitely perform at insufficient levels. All these issues could be solved by adding multiple sensors to the architecture.

Based on the studies from Hall, Elmenreich and Andler, we can group most of the advantages in two categories: *authenticity* and *availability* improvements. The first group refers to those benefits that improve the quality of the measurement, whereas the second one encompasses those benefits regarding ranges or data dimensions. The set of multiple sensors can also be of different types (*heterogeneous*), or of the same type (*homogeneous*), with redundant measurements. There may be benefits that are exclusive to the addition of different sensors, others exclusive to the addition of redundant sensors and those that can happen both ways. Figure 2.2 presents a schematic with the different advantages expected in sensor fusion.
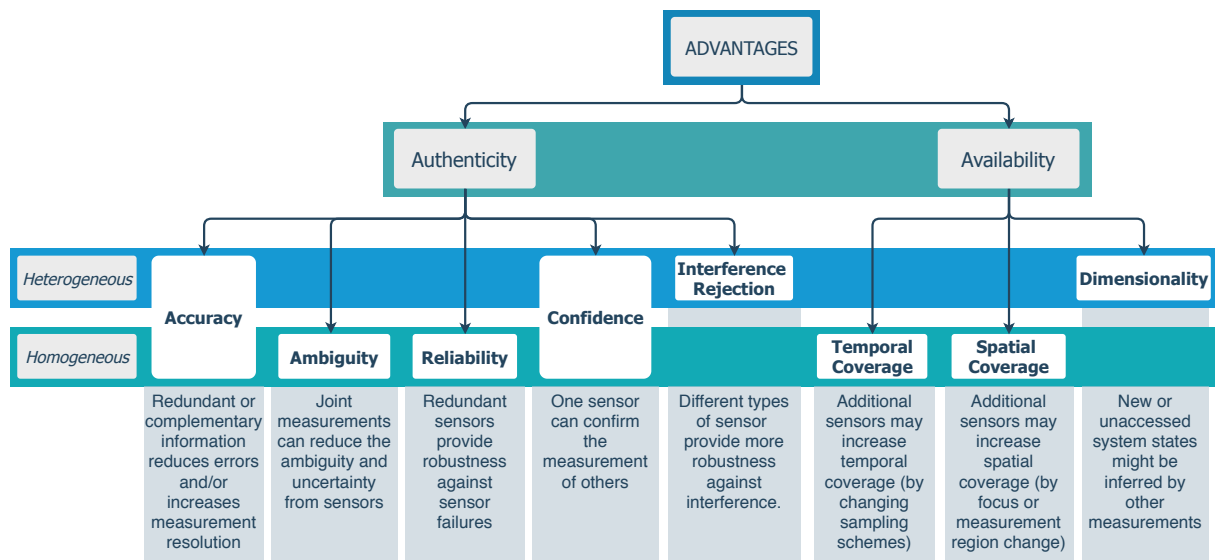
Figure 2.2: Sensor fusion categorization hierarchy based on expected advantages. The white boxes represent the advantages grouped two ways: between autenthicity or availability improvements; and between heterogeneous, homogeneous or both sensor architectures

2.3   Taxonomy and Classification

Sensor fusion definitions have evolved throughout the years. The work of (Boström et al., 2007) analyses more than 30 papers on this matter, to propose a more comprehensive and precise definition to the broad area of information, data and sensor fusion, which we reproduce here:

"Information fusion is the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making."

Researchers in the field made additional efforts to categorize the fusion techniques, using different approaches. One of the earliest attempts comes from (Durrant-Whyte, 1988), where he considered the dynamic use of information in the fusion processes, creating the so-called dependence model, which grouped sensor fusion in three categories: *competitive*, *complementary* and *cooperative*. Competitive type occurs when multiple sensors measure the same properties, usually referred to as redundant architecture. Complementary fusion describes the scheme of different types of sensors measuring different information about the same global object or feature, enabling a more complete fused information, like multiple cameras covering a large area. And the last category, cooperative fusion, happens when more complex data are combined to provide information that would not be available (or hard to obtain) otherwise. An example would be multiple measurements being processed to create soft sensors. An illustrative schematic is presented in Figure 2.3. An abstraction from the human sensory system to Whyte's model can be done by understanding the way flavors are perceived by our taste and smell sensors, tongue and nose, respectively, in a cooperative fashion, while our both eyes or our both ears perform some sort of complementary fusion.

Another common way to categorize sensor fusion is by the *three-level hierarchy* based on input and output characteristics, which depends on the processing stage at which information is fused. The lower-level is related to *raw-data fusion*, where signals from sensors are combined. The mid-level is usually related to *feature fusion*, where information about characteristics of the object are used in the process. The higher level involves *decision fusion*, that can be understood as a reasoning process, like the methods of evidential belief or fuzzy logic. (Dasarathy, 1997) extended this terminology, proposing five fusion modes, according to Figure 2.4. *Data in - data out fusion* (DAI-DAO), the lowest level fusion, processes raw data and outputs raw data, but with some improvements, such as increased accuracy. *Data in - feature out fusion* (DAI-FEO) extracts features from raw data to describe characteristics of the measured environment. *Feature in - feature out* (FEI-FEO) aims at the refinement of the features entering the process, similarly to what DAI - DAO does to raw data. *Feature in - decison*
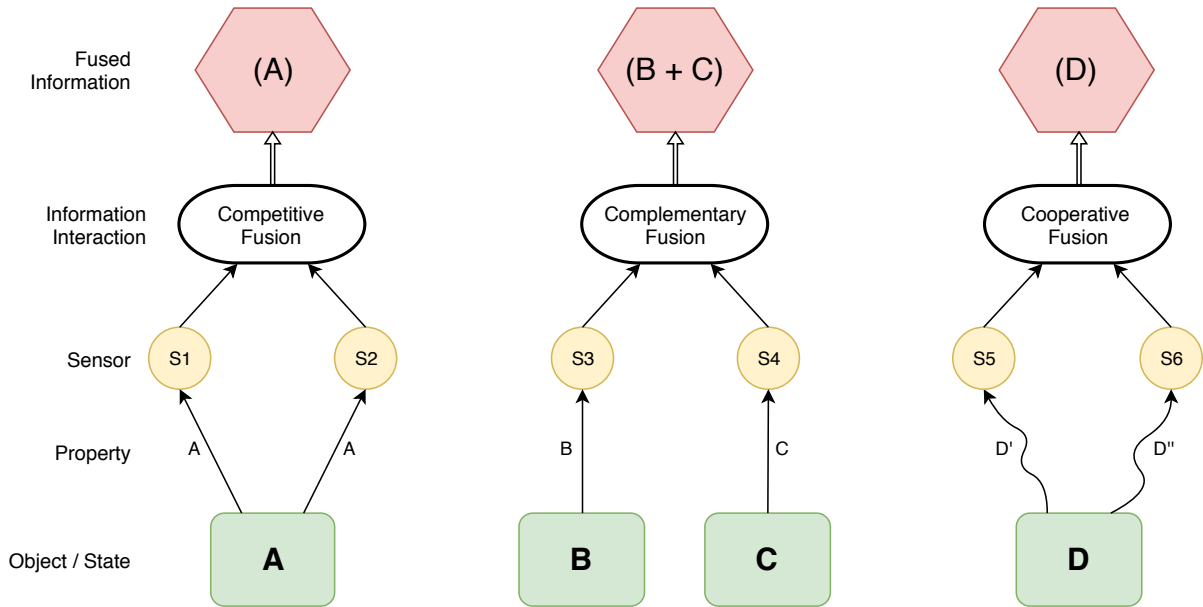
Figure 2.3: Classification of data fusion based on sensor interaction. The lower green boxes represent the states or objects being measured. Their properties can be observed directly (straight lines) or indirectly (curved lines) by sensors indicated by the letter S. After fusion, the output information is presented in the red hexagons. Adapted from (Elmenreich, 2002).

out (FEI-DEO) usually performs classification based on a set of features received as inputs. *Decision in - decision out* (DEI-DEO) outputs a better global decision based on local, restricted decisions. Using our human brain data fusion analogy, many examples can be framed into Dasarathy's terminology. The processing of raw signals, such as letters symbols, into features such as words and texts can be interpreted as DAI-FEO fusion. On the other hand, the process of assimilating features of objects from our eyes and ears, and fusing them into a decision about what they are, for instance, can be perceived as an example of FEI-DEO fusion.

The work of (Castanedo, 2013) provides a comprehensive review of these and other classifications of data fusion techniques. His efforts went beyond as he proposed an interesting new approach, based on the type of architecture, summarized in Figure 2.5. In his model, fusion techniques that collect all measurements in a central processor lie on the *centralized* architecture category. Assuming perfect data alignment and association, such scheme should be optimal. However it is keen to many sampling irregularities issues and might provoke network congestion. When a network of nodes is used, each with its own processing capability, the architecture becomes *decentralized*. Such modular
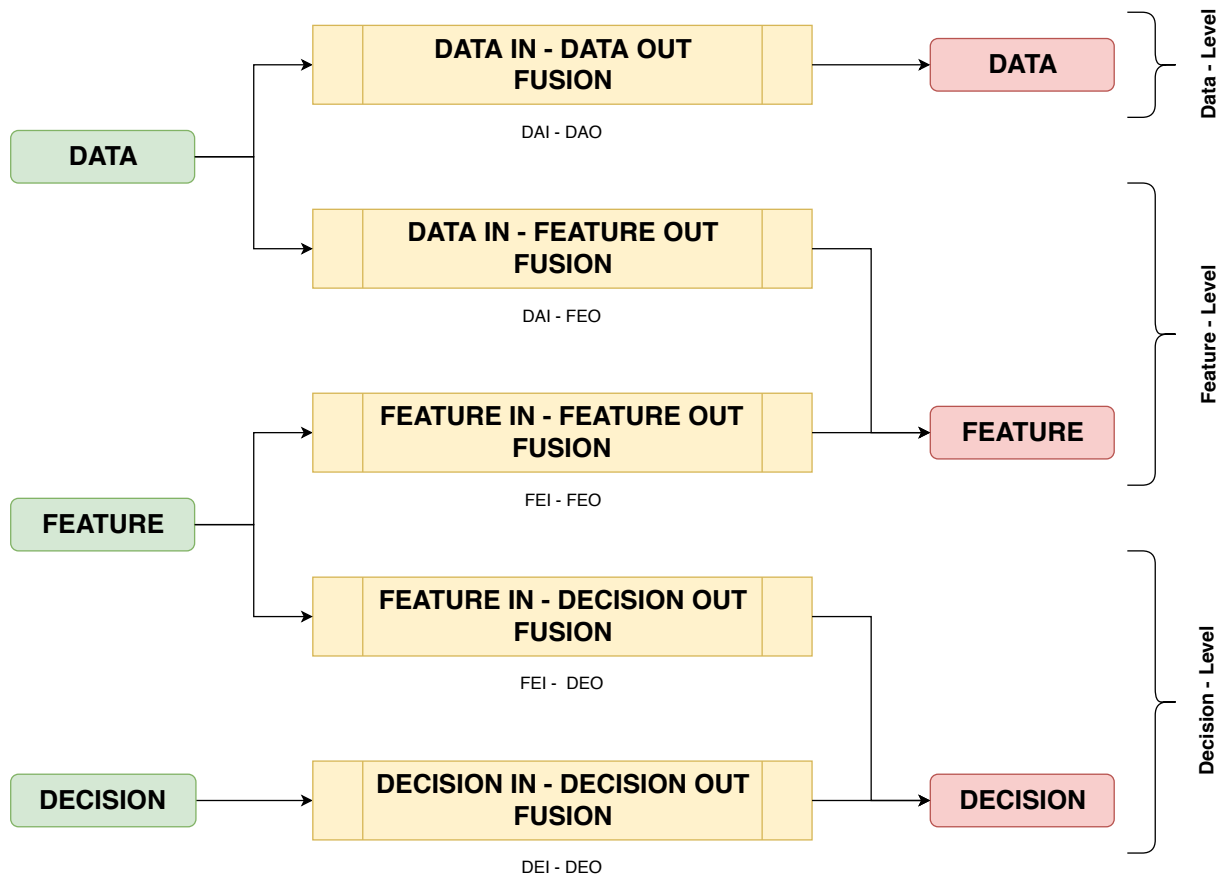
Figure 2.4: Input and output model categorization, representing type of input in green, the method in yellow and the output in red. The fusion levels are shown in the curly brackets.

strategy ensures scalability, since there are no limits to centralized bottlenecks, and survivability to the loss of a particular sensing node. However it can greatly increase communication costs. The third and last configuration is the *distributed* architecture, where each data association is performed by local nodes. The separate outputs are then transmitted to a fusion node, that processes these locally obtained estimates to produce a fused global estimate. This scheme will reduce both the communication costs from decentralized architecture and computational costs from the centralized one, while lacking some of their benefits. A fourth architecture could be defined as *hierarchical*, which combines decentralized and distributed schemes, performing fusion at multiple levels. Getting back to our human sensory capacity, a very complex hierarchical architecture would best describe our brain fusion scheme in Castanedo's classification.
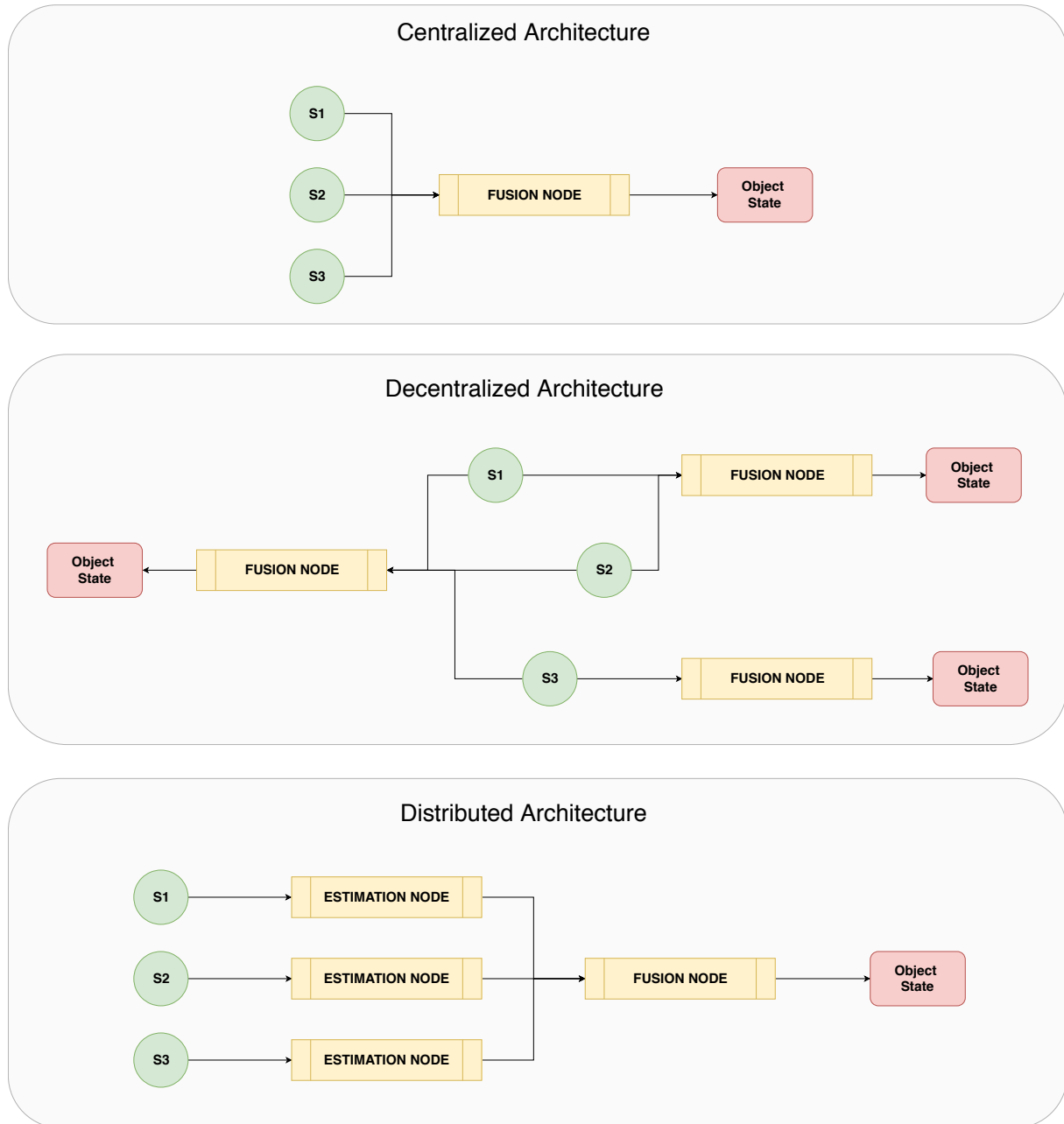
Figure 2.5: Sensor network architectures for data fusion: *centralized*, *decentralized* and *distributed*. Sensors are shown in green circles and the arrows represent the information flow.

A final note on the taxonomy of data fusion methodologies will be given to the work of (Khaleghi et al., 2013), due to its connections to sensor fusion in the presence of irregularities, like the ones that will be discussed in Chapter 3. The idea was to study the methods based on data characteristics that make data fusion a challenging task. The authors referred to these aspects of data as data-related challenges and

categorized the methods by which challenges the methods address. An overview of the challenge hierarchy proposed by the authors is presented in Figure 2.6, with four main categories of how challenging input data can be: *imperfect*, *correlated*, *inconsistent* a0nd *disparate*. The most fundamental problem present in data is imperfection. Indeed, most of the algorithms framed in the other three categories are basically methods that try to neutralize, avoid or minimize their aspects, so that imperfection is the only thing left on data. When correlation is present on data, for instance, the requirements for typical fusion algorithms, such as the Kalman Filter, are broken, so there are methods to eliminate correlation or to minimize its effects, given certain assumptions. In case of data inconsistency, due to outliers, one can act on the sensor outputs directly, to validate information or to detect and remove them automatically. If there are out-of-sequence measurements (OOSM) in data, a type of inconsistency, then the usual frameworks would be: to ignore; to reprocess or use backward/forward prediction; and to augment the state matrices in order to incorporate delayed measurements. Conflicted and disparate data are more specific and are beyond the scope of this study.
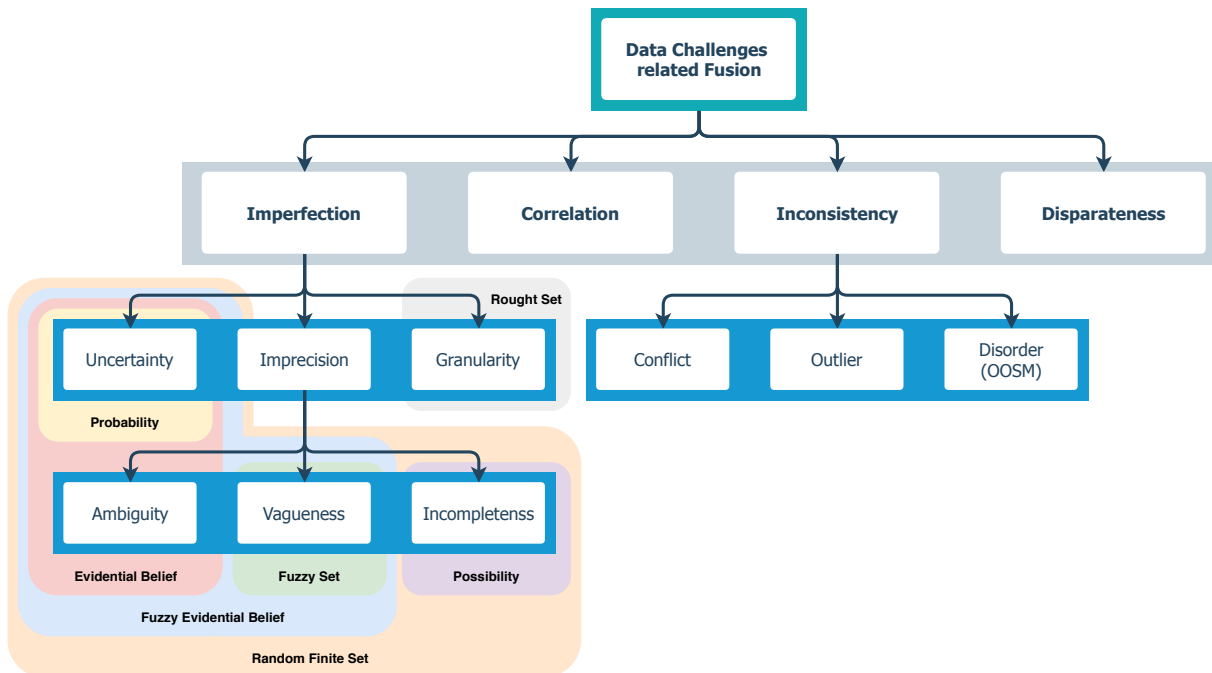
Figure 2.6: Categorization based on data challenges hierarchy. For imperfect data, fusion methods and the issues they deal with are also presented.

2.4   Approaches for the Fusion of Imperfect Data

Being the most fundamental and common challenge present on data, imperfection is also the main focus for research in the area. Based on Khaleghi's classification, imperfection on data can be manifested as uncertainty, imprecision and granularity. We can distinguish uncertainty and imprecision with two information examples: *I believe Maria is one point eight meter tall*; *I am sure that Maria is tall*. In the first sentence, the information is precise, but uncertain. In the second, the information about Maria's height is certain, but imprecise, with some vagueness (fuzziness) attached to it. Usually the amount of precision in data is inversely proportional to the level of certainty. urce of imprecision can also be ambiguity, as in the phrase: *Maria gave birth to her daughter yesterday at 5*. We don't know if it was 5 PM or 5 AM, though the information was certain. The last type of imprecision would be for incomplete data, when we have missing information. The sentence "*Maria's height is above one point five meter*", is incomplete, for example, meaning that only one bound was given. Any height above one hundred and fifty centimeters is possible and any height less than or equal to it is impossible, defining the so-called possibility measures. Finally, granularity is an imprecision related to the internal structure of data, referring to the capacity of distinction among states. Different attributes on the data or a different set of possible states will generate different levels of imprecise information.

Given the amount of potential problems in data and their particularities, it is only natural that no data fusion approach alone could tackle all of them. Researchers have proposed various approaches that focus on one or a few of these issues and (Khaleghi et al., 2013) explore methods for all the challenges in their categorization model. In this study we will limit ourselves to imperfect data, for which, in most cases, the mathematical background of the algorithms relies on the representation of imperfection. These methods are highlighted in Figure 2.6 in different colors and covering one or multiple aspects. In the next subsections we will explore them. Table 2.3 presents a summary of all methods with their main advantages and limitations, adapted from the study of Khaleghi and his coauthors.

2.4.1 Probabilistic

Uncertainty is the most natural source of data imperfection and it is usually modeled by probability density functions (PDFs). Thus, probabilistic fusion methods are the most adequate to handle it. The most classical approach to fuse data based on uncertain measurements is using Bayes' theorem. The idea is to update the probabilities of an hypothesis H given some evidence E by using his famous equation (Stone, 2013)

$$\rho(H|E) = \frac{\rho(E|H)\rho(H)}{\rho(E)}, \tag{2.1}$$

where $\rho(H|E)$ represents a conditional probability, that is the probability of H being true, given E. We can interpret (2.1) as a fusion of *a-priori* belief of an hypothesis ($\rho(H)$) and some current evidence ($\rho(E|H)/\rho(E)$) to come up with a better current estimate of the hypothesis ($\rho(H|E)$).

In addition to Bayes' contribution, Gauss' least squares methods and Fisher's maximum likelihood estimation are the foundations of all probabilistic approaches to data fusion, recursive filtering and state estimation. (Kolmogorov, 1962) in 1941 and (Wiener, 1949) in 1942[1] independently designed a linear minimum mean-square estimation technique, that is considered to be the first probabilistic designed filter, the so called Wiener-Kolmogorov filter. A few years later, perfecting the work of its predecessors, (Kalman, 1960) developed the recursive mean-square filter, a groundbreaking method known as Kalman Filter (KF). Its impact was so big, that only one year after its publication, Kalman's algorithm was used in the Apollo Project (Mohinder and Angus, 2010), to solve its guidance and navigation problem. A review on the evolution of least-squares estimation theory is provided by (Sorenson, 1970).

When it comes to a multi-sensor system, (Willner et al., 1976) introduced three approaches for the discrete Kalman Filter: parallel, sequential and data compression filters. In the parallel filter, all measurements are processed by a KF in parallel, producing a multi-output estimation. For the sequential filter design, multiple KFs are used, where the estimates of a predecessor KF are used as input for the successor KF. Data compression or output fusion filter design compresses similar data using their noise covariance matrix beforehand and the fused output is used as the measurement for a single KF. A fourth method was proposed by (Singer and Kanyuck, 1971), referred to

---

[1]Their discoveries went public a few years later due to secrecy during war times.

as track-to-track fusion (TTF), that employs single-output KFs and fuses their outputs considering the correlation between them.

2.4.2 Evidential Belief

A different framework for managing imperfection is based on Dempster-Shafer evidence theory (DSET) (Shafer, 1976). Shafer argued that his theory, extended from Dempster findings, includes the Bayesian approach as a special case of evidence combination or information fusion. The difference between both relies on the assignment of uncertainty. Whereas Bayesian framework considers one multi-variable PDF to represent each state's uncertainty and the respective covariances, evidential belief theory assigns uncertainties not just to the states, but also to all its possible subsets, using probability mass functions. Imagine the example of Maria giving birth and the information about the time it happened. We can assign uncertainties to the ambiguous possibilities "5 PM" and "5 AM", in order to use the data. If the uncertainty about the information was also an aspect of the data, evidential belief approach could handle it likewise.

Therefore, DSET can be more adequate when fusion takes place at a higher-level estimation, that is the decision-level, referring back to the input and output classification model from Figure 2.4. Situations related to risk assessment configure classical applications (Srivastava, 2011), where we have features or local decisions as inputs, usually with ambiguous and conflicting information and we need to take a global decision out of it.

2.4.3 Fuzzy Logic

Fuzzy logic, first proposed by (Zadeh, 1965) is ideal to handle vagueness in information. Unlike classical crisp definitions, where an element x belongs or not to some definition A, fuzzy sets are characterized by a *membership function* $\mu_A(x)$ which associates a degree of pertinence of x in the interval $[0, 1]$ in A. That is, the closer $\mu_A(x)$ is to the unity, the higher is the grade of membership of x in A. Instead of defining tall people as those with height above certain crisp lower bound, we can define a membership function that assigns continuous degrees of "tallness" to different people. Zadeh also generalized the crisp set notion of operations to the fuzzy set theory: fuzzy complements (NOT), fuzzy intersections (AND) and fuzzy unions (OR).

Vague data can be then fused using fuzzy inference systems (FIS), mapping input variables into an output space using fuzzy logic operators through a set of If-Then fuzzy rules. wo of the most widely used FIS are the Mamdani-type (Mamdani and Assilian, 1975) and Sugeno-type (Sugeno, 1985). The main difference between both is that the outputs of the former rules are also a fuzzy set, while the outputs of the latter rules are polynomials with respect to the input variables. Mamdani and Sugeno-type rules are given, respectively, by

$$R_i^{Mamdani} : \text{If } x_1 \text{ is } A_{i,1} \text{ and } x_2 \text{ is } A_{i,2} \text{ ... and } x_n \text{ is } A_{i,n}, \text{ Then } y_i \text{ is } C_i, \tag{2.2}$$

$$R_i^{Sugeno} : \text{If } x_1 \text{ is } A_{i,1} \text{ and } x_2 \text{ is } A_{i,2} \text{ ... and } x_n \text{ is } A_{i,n}, \text{ Then } y_i \text{ is } f_i(X), \tag{2.3}$$

where $x_j \in \mathbb{R}$, $\forall j = 1,...,n$ is the jth input variable, $A_{i,j}$ is the jth antecedent fuzzy set for the ith rule, C_i is the consequent fuzzy set for the ith rule, $f_i(X)$ is the output polynomial with respect the the input vector X.

Therefore, we can understand the inference process as a series of steps, according to Figure 2.7. First, we fuzzify the inputs via membership functions. Then we apply the logic operators defined by each rule and find the result implication on the consequent. Finally we aggregate the consequents accross all rules and defuzzify its result.

Fuzzy set theory differentiates itself from probabilistic and evidential reasoning theories by modeling the fuzzy membership of a state whose class is ill-defined, whereas the other methods model uncertainties in well-defined state classes.

It is possible to combine fuzzy theory with DSET to handle the imperfections that both approaches can deal with, that is uncertainty, ambiguity and vagueness all together, using the fuzzy evidential belief framework.

2.4.4 Possibilistic

Thirteen years after developing the mathematical background of fuzzy information, Zadeh introduced the concept of possibility theory (Zadeh, 1978), using fuzzy sets as its basis. According to him, fuzzy sets are to possibility theory what measures are to probability.

A membership function $\mu_A(x)$ of a fuzzy set A of a universe of discourse X can be interpreted as the compatibility of x with the concept labeled as A. Letting U be a variable that takes values in X, we can define the *possibility function* π_x of x associated
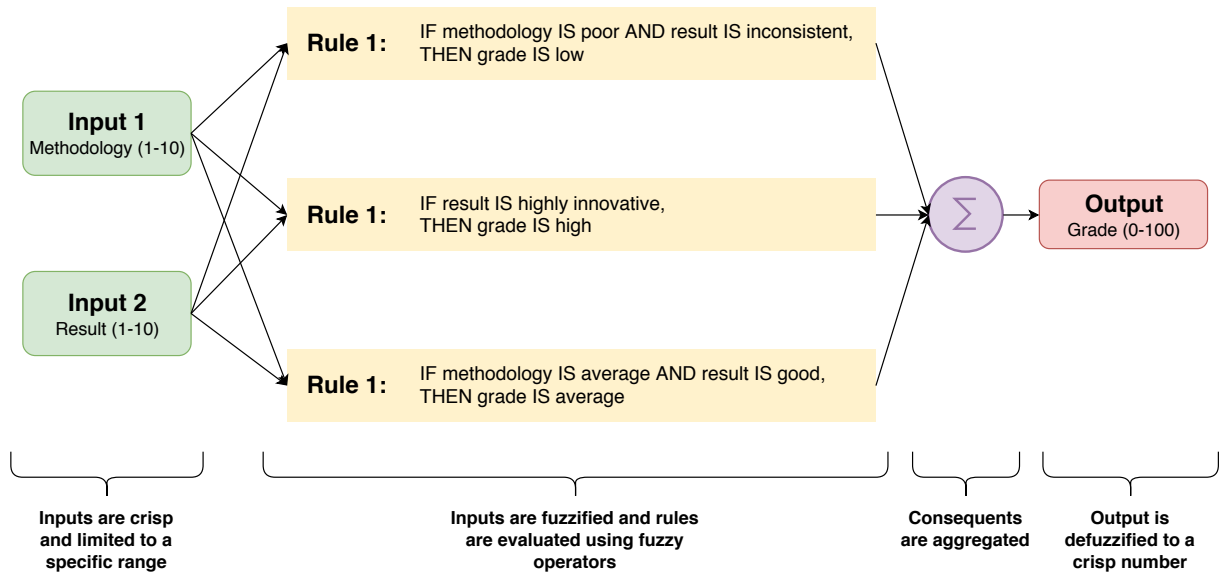
Figure 2.7: Fuzzy inference system example for grading thesis. There are two inputs: methodology and result. They are fuzzified according to the antecedent membership fuzzy sets: poor and average for methodology; and inconsistent, consistent and highly innovative for result. After evaluation of the rules, the implication on the consequents are calculated, producing output fuzzy sets, that are finally aggregated. The final output is then defuzzified to produce a crisp number for the grade, between 0 and 100.

with U to be equal to the membership function of A, that is $\pi_X(u) \triangleq \mu_A(x)$. The interpretation, however, is that the closer $\pi_X(u)$ is to the unity, the more plausible that value is to be true.

Another way to understand possibility functions is in its comparison to density functions in probability. In the information "Maria has failed a few times in the Digital Control course", let us consider X as the number of times in the universe $U = 1,2,3....$ The membership function $\mu_A(x)$ can model how close to "a few times" is the fuzzy variable x. From such model, we also define the possibility function $\pi_X(u)$ that can be interpreted as how possible it is that Maria has failed u times in that course. We could also model such lack of information in data by a probability function $\rho_X(u)$, as in how likely it is for Maria to fail u times. Let us consider that an educated set of criteria was employed to define the discrete values of both functions as shown in Table 2.1

Note that according to Table 2.1 it is perfectly possible for Maria to have failed 1, 2, 3 of 4 times. The degree of possibility decreases for higher values and there is an upper bound, from which Maria would have been expelled. On the other hand, based on recent flunk history, the most probable number of times for Maria to have failed

Table 2.1: Possibility and probability functions associated with how many times Maria has failed Digital Control

u	1	2	3	4	5	6	7	8	9
$\pi_X(u)$	1	1	1	1	0.8	0.6	0.4	0.2	0.1
$\rho_X(u)$	0.3	0.4	0.2	0.05	0.03	0.02	0	0	0

is 2, whereas since no student failed more than 6 times, the probability that Maria will fail that many is 0. Thus, possibilistic approach can be more appropriate to cope with incomplete data, with missing information about a lower or an upper bound, for instance, in which case it is clear that some values are impossible instead of unlikely.

The fusion approach to possibilistic data models was extensively studied by (Dubois and Prade, 2000) and it is similar to the rules employed in fuzzy fusion. The design of the rules set is based on how plausible the sources of data are.

2.4.5   Random Set

So far, the presented methods are able to cover all aspects of uncertainty and imprecision. Although some of them can tackle multiple aspects, like DSET and Fuzzy DSET, none of them can handle all the sources of imprecision and uncertainty altogether. The random set approach to data fusion, proposed by (Goodman et al., 1997), offers such potential to integrate all these aspects in one unifying structure. The idea was to generalize the single-sensor, single-target statistics (random variables) to a broader multi-sensor, multi-target statistics (random sets), also known as finite-set statistics (FISST). The direct mathematical parallels between them are presented in Table 2.2.

By modeling the system states and measurements as random sets of finite size instead of vectors of random variables, a variety of different phenomena can be described, such as target disappearance or appearance, extended or unresolved targets, missing measurements and false alarms (Khaleghi et al., 2013). As described by (Goodman et al., 1997), the random set approach can model systems that are comprised of randomly varying numbers of randomly varying objects of various kinds.

Mathematically speaking, random sets are random elements whose values are sets. A random set U is a finite set, whose power set $\mathcal{P}(U)$ is composed of elements described by some specific probability law. In other words, it is defined by

Table 2.2: Direct mathematical parallels between single-sensor, single-target and multi-sensor, multi-target. Adapted from (Goodman et al., 1997; Mahler, 2004)

Single-sensor / target	Multi-sensor / target
random vector, Z	finite random set, Σ
sensor	global sensor
target	global target
observation, z	global observation-set, Z
parameter, θ	global parameter-set, Θ
derivative	set derivative,
integral	set integral,
probability measure	belief measure,
pior PDF	global PDF,
likelihood	global likelihood,
information theory	multi-target information theory
filtering theory	multi-target filtering theory

$$f : \mathcal{P}(U) \to [0,1] \quad \text{with} \quad \sum_{A \in \mathcal{P}(U)} f(A) = 1 \tag{2.4}$$

where f can be interpreted as a PDF defined on sets rather than on points of U. That is, the probability that the subset A of U is selected is $f(A)$.

Efficient applications of random set theory have been studied in tasks such as system identification and time-series forecasting (Nuñez-Garcia and Wolkenhauer, 2002), target tracking (Maehlisch et al., 2006) and econometrics (Molchanov and Molinari, 2014).

2.4.6   Rough Set

Granularity is the only type of imperfection left from the categorization presented in Figure 2.6. It refers to the extent to which objects can be distinguished by data, considering the features or attributes that define them. Additionally, the way a set of features is designed will depend on a given knowledge base. Some objects might be discernible considering one set, but indiscernible in another set. If we choose to characterize a group of people by their age and height, a woman and a man might be

indiscernible. If we add sex to the set of features, they become discernible. For the sensor fusion field, we can think of data being collected in a very refined universe of discourse, while the universe of concepts in which we transform data into knowledge is coarser, and thus some objects in the data might be indiscernible. These indiscernible objects in rought set theory are referred to as *elementary sets* or *elementary granules* for a specific set of features. The union of these *elementary sets* forms what is called *definable sets*.

Based on these concepts, (Pawlak, 1991) developed the rough set theory, which enables dealing with different data granularities by means of *approximation spaces*. The idea is to provide crisp lower and upper bounds to undefined sets of objects in a given knowledge base. Let B be a subset of features chosen to describe objects in the universe of A, that is $B \subseteq A$. If there is a target set X that is undefined in B, that is there are objects x in X that are indiscernible by A, its definition can be approximated by two sets (Pawlak and Skowron, 2007)

$$
\begin{aligned}
B_*(X) &= \{x \in A : B(x) \subseteq X\}, \\
B^*(X) &= \{x \in A : B(x) \cap X \neq \emptyset\},
\end{aligned}
\tag{2.5}
$$

and

$$
BN_B(X) = B^*(X) - B_*(X),
\tag{2.6}
$$

where x represents elements or objects in the universe of A, $B_*(X)$ and $B^*(X)$ are the *B-lower* and *B-upper* approximation of X, respectively and $BN_B(X)$ is the approximated X, also called as *B-boundary* region. If the boundary region is empty, then the set X is crisp or exact with respect to B, whereas if it is not empty, then the set X is rough with respect to B. Figure 2.8 illustrates an approximate space in which the universe is partitioned into elementary squares for which subset X is undefined, but can be approximated is by upper and lower bounds.

In (Pawlak, 1991) and in the references therein, many real life applications of rough set theory are explored, such as civil engineering, medical data analysis, aircraft pilot performance evaluation, vibration analysis and image processing.
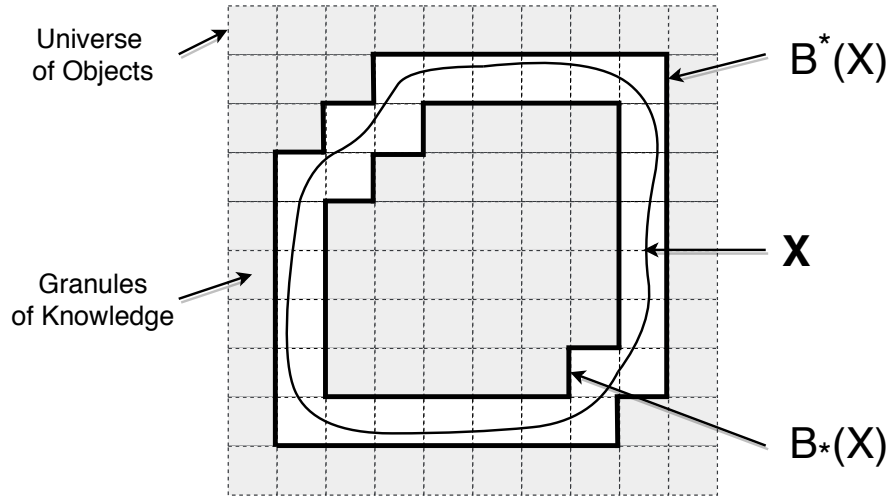
Figure 2.8: Best approximation of the rough set X by lower $B_*(X)$ and upper $B^*(X)$ crisp sets. The elementary granules are squares and the union of these granules form the universe of objects.

2.5   Chapter Summary and Final Remarks

In this chapter, sensor fusion literature is reviewed. We explore the discussion about performance improvement from the combination of information from multiple sources, that has been drawing attention for many decades. The main reasons for the evolution of sensor fusion as a field of science are presented in a comprehensive way, divided by the expected advantages in data authenticity and data availability. We continue with the definition and classification of sensor fusion approaches. Four main taxonomies are reviewed: classification based on sensor interaction; the input/output model based on the three fusion levels approach; sensor network architecture designs; and the categorization based on data challenge hierarchy. The last approach is especially interesting, since it relates to the core of this study: sensor fusion in the presence of irregularities in data. Algorithms that deal with the many aspects of data imperfection are summarized in Table 2.3, such as: probabilistic; evidential belief; fuzzy logic; possibilistic; random set; and rough set theory. Hybrid methods have also being studied in the field, with interesting results.

Being the most popular method in the fusion community, the remaining of our study will focus on probabilistic methods for data fusion and its application on state estimation for sampled-data systems. A new data challenge is often present when complex sensor architectures are used to observe system states: sampling irregularity.

Table 2.3: Data fusion methods for imperfect data, adapted from (Khaleghi et al., 2013, page 35, Table 1)

Algorithm	Approach	Advantages	Limitations
Probabilistic	Bayesian framework to fuse uncertain data represented by probability density functions	Well-established and optimal for certain conditions	Might be unsuited for other data imperfections
Evidential Belief	Data fusion based on probability mass function, using Dempster-Shafer theory and combination rules	Enables fusion of both uncertain and ambiguous information	Incapable of dealing with other aspects of imperfection
Fuzzy Reasoning	Vague data represented by fuzzy set theory and fusion based on fuzzy rules	Intuitive and interpretable approach for vague data, such as human generated	Only applicable to vague data
Possibilistic	Data fusion based on fuzzy theory, with data representation similar to probabilistic and evidential belief	Indicated for poorly informed environment with incomplete data	Not very common and well-established
Rough Set	Imprecise data is approximated based on granularity and manipulated via classical set theory	Dispenses preliminary or additional information	Data granularity must be adequate
Random Set	Extension of Bayesian filter, representing the state space as a random set to capture many aspects of imperfection	Can potentially provide a unified framework for fusion of imperfect data	Not very appreciated by the fusion community
Hybridization	Combination of different fusion methods and data representation	More comprehensive treatment of data imperfection and benefits from complementary fusion	Computational expensive and very problem specific

Irregular Sampling

In the last chapter, we reviewed the main motivations and advantages behind the sensor fusion field of science, as well as its techniques. Despite all the growth and benefits from fusing data form multiple sensors, some challenges naturally appear. For the state estimation problem for sampled-data systems, a common challenge is related to sampling irregularities introduced in the network.

In this chapter, we review the irregular sampling problem. First, the concept of irregular sampling adopted in this study is discussed. In Section 3.2 we categorize the different types of irregularities that may occur in sampling and discuss their main causes and characteristics. Then, in Secton 3.3, each irregularity is further discussed, with examples, mathematical models and their subdivisions, where applicable. We end this chapter with a discussion of time synchronization in Section 3.4, which is needed to guarantee a common time scale for all nodes in network, enabling the irregularities to be dealt with appropriately.

3.1 Definition of Irregular Sampling

Most of the systems studied in estimation and control theories take place in an analog environment with dynamics evolving according to continous-time differential equations. However, due to the many benefits of digital technology implementations, their signal must be sampled in order to be processed, giving rise to the so-called *sampled-data systems*.

In practice, the sampling process is modeled by a sampler and a data hold device, that enables signal reconstruction. The most common data hold configuration is the zero-order holder (ZOH), that locks the signal value at the sampling instant until the next sample is available. Therefore, the output of a sampler that transmits data every T seconds and a ZOH is given by (Phillips and Nagle, 1995)

$$\bar{f}(t) = f(0)[u(t) - u(t-T)] + f(T)[u(t) - u(t-2T)] + f(2T)[u(t-2T) - u(t-3T)] + \dots, \quad (3.1)$$

where $\bar{f}(t)$ represents the reconstructed sampled signal, $f(t)$ is the original continuous-time signal and $u(t)$ is the unit-step function. Taking the Laplace transform of (3.1) we have

$$\bar{F}(s) = \mathcal{L}[f(t)] = f(0)\left[\frac{1}{s} - \frac{e^{-Ts}}{s}\right] + f(T)\left[\frac{e^{-Ts}}{s} - \frac{e^{-2Ts}}{s}\right] + f(2T)\left[\frac{e^{-2Ts}}{s} - \frac{e^{-3Ts}}{s}\right] + \dots$$

$$= \left[\sum_{n=0}^{\infty} f(nT)e^{-nTs}\right]\left[\frac{1 - e^{-Ts}}{s}\right]. \quad (3.2)$$

Only the first term of (3.2) is dependent on $f(t)$ and thus can be interpreted as the sampler, whereas the second term represents the ZOH. Note that neither exists alone, since the real physical signal is composed by both terms. Taking the inverse Laplace transform of the sampler term yields

$$\mathcal{L}^{-1}\left[\sum_{n=0}^{\infty} f(nT)e^{-nTs}\right] = f(0)\delta(t) + f(T)\delta(t-T) + f(2T)\delta(t-2T) + \dots, \quad (3.3)$$

where $\delta(t)$ is the unit impulse function, also known as Dirac delta function. Although it does not represent a real signal, (3.3) is referred to as the *ideal sampler* (Phillips and Nagle, 1995).

Thus if the state observations of a sampled-data system are transmitted through an ideal sampler, it can be modeled as

$$y(t) = g(x(kT), v(kT), kT), \quad \text{for } kT \leq t < (k+1)T, \quad (3.4)$$

where $g\colon \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^+ \to \mathbb{R}^m$ represents the observation model, $x(kT) \in \mathbb{R}^n$ is the state vector, $v(kT) \in \mathbb{N}^m$ is the measurement error, $k \in \mathbb{N}$ is the discrete-time index and $T \in \mathbb{N}^+$ is the sampling interval.

Therefore, a sampled-data system is regularly sampled if its observation model can be modeled by (3.4), as a consequence of an *ideal sampler*. In other words, in this study we refer to *regular sampling* as measurements being taken periodically, with single-rate and transmitted without time-delay and any loss of information. Anything else will be

framed as *irregular sampling*.

3.2   Contextualization

Sampling irregularities may occur due to a variety of issues. Sometimes they occur as undesired side effects of using large sensor networks architectures and others due to deliberate non-uniform sampling schemes. In this section we try to categorize and review the main irregularities observed in practice. The diagram in Figure 3.1 provides a simplified overview of them, separated by their sources.

Networked system monitoring and control appears to be the main cause of irregular sampling. Unreliable communication channels may lead to random time delays and loss of information, specially if data are transmitted using a common media (Sahebsara et al., 2007; Moayedi et al., 2011). In case they get randomly interrupted during transmission or if a sensor fails at some point, the signal received may predominantly contain noise, causing uncertain observation or packet dropouts (Hadidi and Schwartz, 1979; Wang et al., 2009). Systems that are observed by a large number of desynchronized sensors will provide observations at random time intervals (Micheli and Jordan, 2002). If they are synchronized but designed to operate in a centralized fashion, there is a chance that different time delays are produced due to distinct transmission routes for each sensor (Bar-Shalom, 2000; Challa et al., 2003; Anxi et al., 2005).

However, the communication networks shall not always be held responsible. Some applications are designed to be measured in an irregular way. In event-based schemes, for example, the measurements are transmitted only when certain conditions are met (Liu et al., 2014; Zou et al., 2017). Such approach can reduce communication resource consumption substantially (Hu et al., 2017), but it will cause aperiodic sampling. Non-Uniform Sampling (NUS) is also intentionally used as an alias detection method (Kunoh, 2015) or to enhance the spectral resolution of signals, largely used in Nuclear Magnetic Resonance (NMR) spectroscopy analysis (Hyberts et al., 2013), for instance. In other situations, due to the nature of the process being observed, the measurement strategy relies on different procedures. For a lot of chemical processes, for instance, the variables can be measured in an online, fast rate and delay free fashion, but providing inaccurate data. Therefore, lab analyses are used to improve estimation quality, but usually at slower rates, sometimes irregularly and with possible time delays (Fatehi
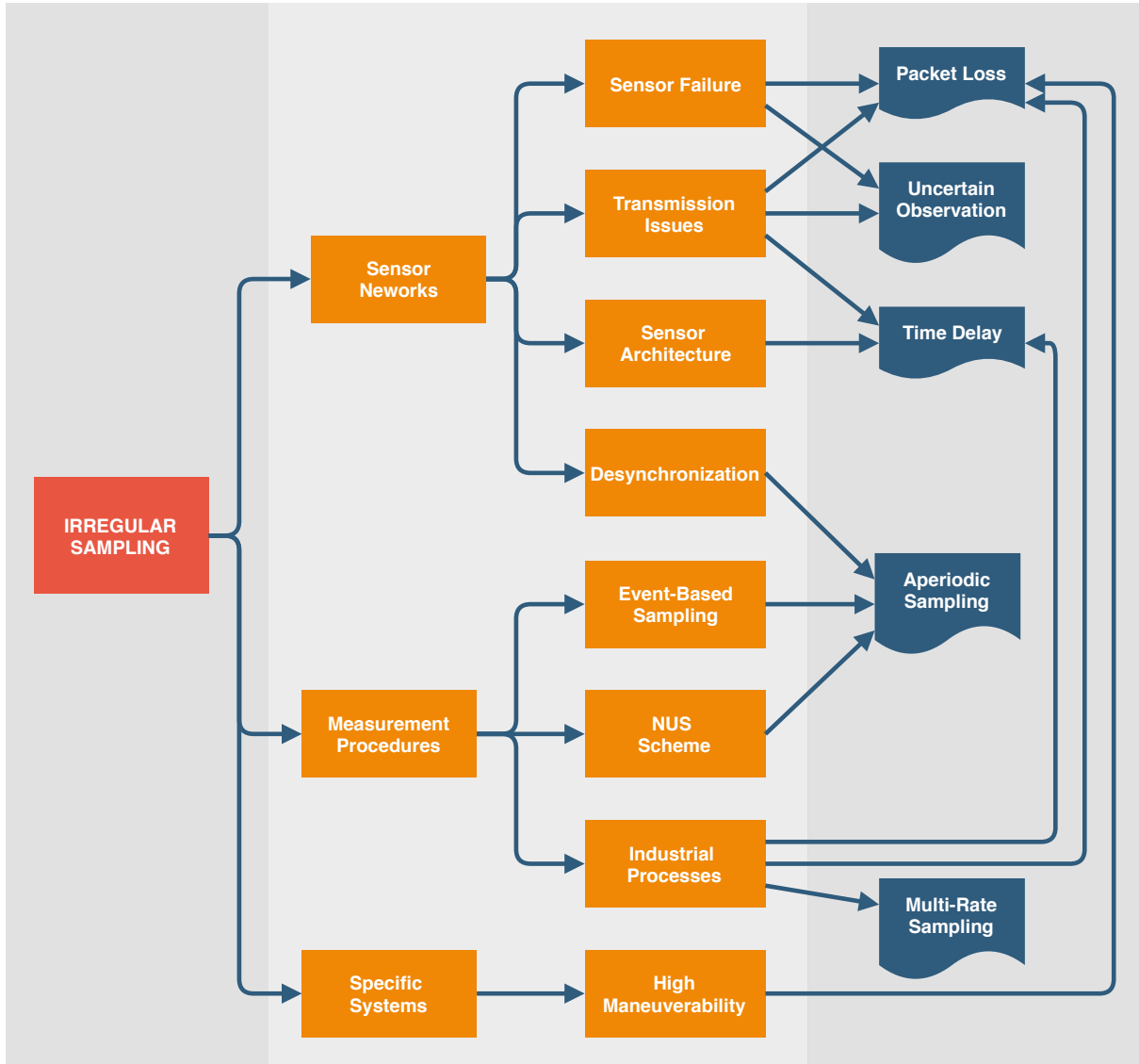
Figure 3.1: Irregular sampling diagram, showing the main causes (in orange) and effects (in blue)

and Huang, 2017). Other industrial applications suffer from the same dilemma, and the sampling scheme ends up with a multi-rate data transmission, with random time delays and possibly measurement scarcity (Peñarrocha et al., 2012).

Finally, sampling irregularities might also appear due to a specific nature of a system. In some high maneuverable target-tracking applications, for example, there is a chance that the sensor misses the target, transmitting only noise, leading to the so called uncertain observation issue (Wang et al., 2009; Chen et al., 2013).

Whatever the reason for the irregularities, data need to be associated in order to

be fused into knowledge. A crucial part of association is temporal synchronization of observations, so that the exact time at which measurements are taken are available (Ping, 2003). Most sensor fusion methods for irregularly sampled systems rely on the correct time-stamps to develop modifications in classical algorithms. If observations are imprecisely time stamped, some alternatives have been proposed (Julier and Uhlmann, 2005; Huck et al., 2011) that incorporate some aspects of that imprecision in the estimation method. Still, some knowledge about the irregularity is assumed to be known. Alternatively, many techniques for time synchronization can be performed, to ensure a common timescale for all the sensors.

On the next sections, we review the main irregular sampling types and the main approaches to deal with time synchronization in sensor networks.

3.3 Types of Sampling Irregularity

3.3.1 Time Delay

Time-delay systems (TDS) are probably the most common mathematical representation to time delays in practice. The works of (Richard, 2003; Fridman, 2014) and the references therein provide a good coverage of the subject. In TDSs, there might be delays in the input or in the output signals, introduced by communication networks, or even in the states themselves. The latter phenomenon is called system with aftereffect or dead-time. Since we are studying the irregular sampling issue, only signal delays are relevant to us. A summary of causes and effects that time delay causes in an estimator are illustrated in Figure 3.2.

Considering delays in the measurement model only, (Lu et al., 2005) studied the estimation problem when they are constant and known. They describe a discrete linear measurement model observed by l different systems with delays as

$$y_i(kT) = H_i(kT)x(t_i) + v_i(kT) \tag{3.5}$$

where k is the discrete-time index of measurements, T is the sampling interval, $i = 0, 1, \ldots, l$ and l is the number of different known delays for each observation system. $y_i(kT) \in \mathbb{R}^{p_i}$ are delayed measurements and $v_i(kT) \in \mathbb{R}^{p_i}$, the measurement noise. The known delayed time instants are given by $t_i = t_{i-1} - d_i$, with $d_0 = 0$, $d_i > 0$ for $i > 0$ and

Figure 3.2: Time delay diagram, showing the main causes (in orange) and effects (in dark blue) of time delay irregularity. The light blue boxes indicate different types that lead to different effects.

$t_0 = kT$. For a given time instant kT, $y_i(kT)$ represents the observation of state $x(t_i)$ at time kT, with total delay $\sum_{k=1}^{i} d_k$. Thus, if $T \geq d_1 + \ldots + d_l$, the complete observation vector of state $x(t_i)$ at time kT, becomes $[y_0^T(kT) \ldots y_l^T(kT)]^T$. On the other hand, if the sampling interval is smaller than the delay of one or more observation system, that is $d_1 + \ldots + d_{i-1} \leq T < d_1 + \ldots + d_l$, then observation of the system is given by $[y_0^T(kT) \ldots y_{i-1}^T(kT) \; 0 \ldots 0]^T$.

For some systems, delays might not be known and constant, but still multiple of a fixed value. In such cases, observations might be received in a burst, when more than one packet arrive between two consecutive sampling instants. When that happens, the estimator might use only the latest measurement and discard all others, or implement a buffer to iterate over all received packets (Moayedi et al., 2011).

However, in many applications the measurements are received by the estimator with irregular and unknown delays, although taken at regular time intervals. In such cases, time delays can be interpreted as a stochastic process $d(k)$, varying randomly throughout time. (Han and Zhang, 2009) describes a discrete-time measurement model for random delayed observations as

$$y(k) = H(k)x(k - d(k)) + L(k)v(k) \tag{3.6}$$

where $d(k) \in \mathbb{N}$ is a random but bounded time delay, assumed to be a discrete-time Markov Chain with finite space $\{0, d\}$, observable at each sampling time k.

Multiple of a known lag or not, delayed measurements from a multisensor system are subject to arrive disordely, which leads to the sampling irregularity commonly known as out-of-sequence-measurements (OOSM). It can be classified in three ways, depending on the number of lags, according to Figure 3.3.
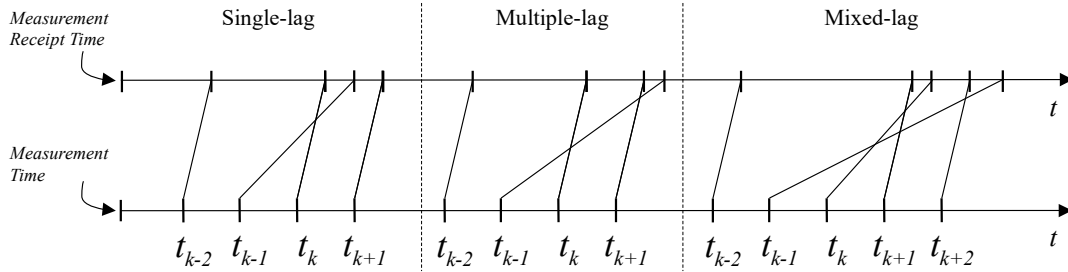
Figure 3.3: Different classes of out-of-sequence measurements irregularities

(Anxi et al., 2005) describes four different filtering approaches to deal with OOSM: reprocessing, that stores filter results to rollback with the time-delayed measurement; data buffering, that holds a set of measurements, greater than the maximum expected lag, to be sorted before filtering; discarding data, that neglects time-delayed measurements; and directly updating, that uses the delayed information to update current state estimate. (Bar-Shalom, 2000) used the last approach to describe an optimal filter for the single-lag case.

3.3.2 Packet Loss

When data are being transmitted by a large network of sensors, there is a probability that they get lost in the way or they might arrive after a significant delay, which is equivalent to a loss for practical applications (Sinopoli et al., 2004). Usually referred to as packet dropout or loss, missing or intermittent observations or scarse measurements (Albertos et al., 2004) they may happen due to node failures, network congestion,

limited bandwidth or temporal failure. A summary of causes and effects that time delay causes in an estimator are illustrated in Figure 3.4.
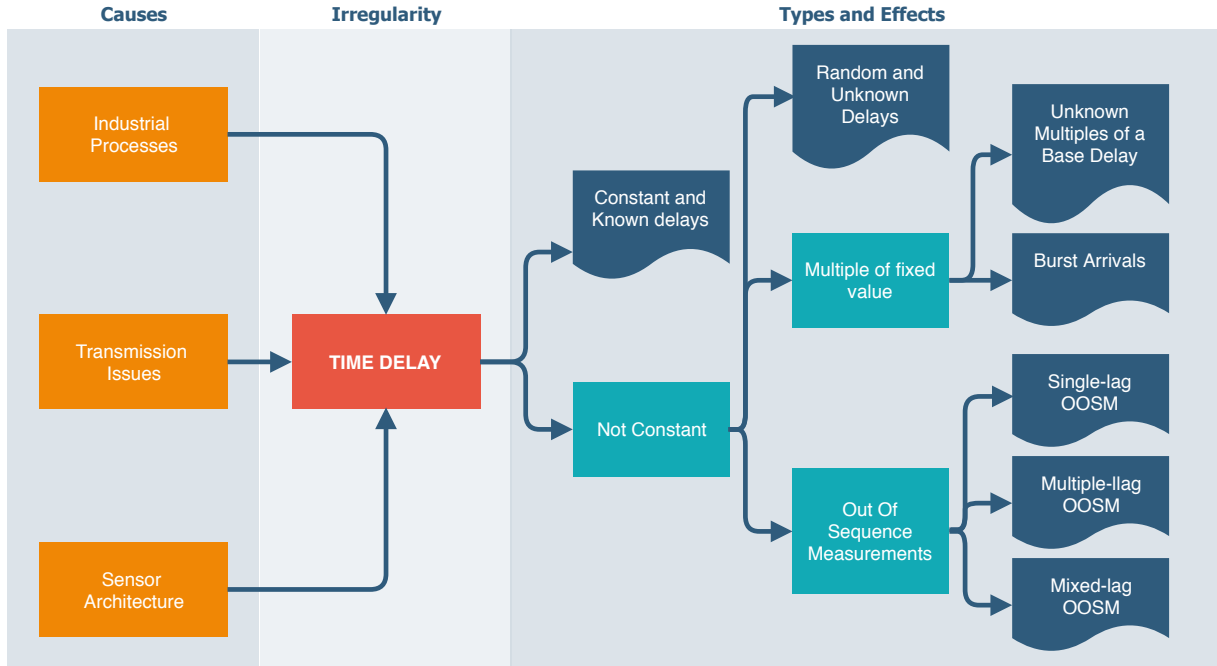

Figure 3.4: Packet loss diagram, showing the main causes (in orange) and effects (in dark blue) of time delay irregularity.

Mathematical description of packet dropouts can be carried out recursively, as described in (Ma and Sun, 2011), by

$$
\begin{aligned}
z(k) &= H(k)x(k) + v(k), \\
y(k) &= \xi(k)z(k) + (1 - \xi(k))y(k-1),
\end{aligned}
\tag{3.7}
$$

where $z(k) \in \mathbb{R}^m$ is the measured output transmitted to the estimator, $v(k) \in (R)^m$ is white noise, $y(k) \in \mathbb{R}^m$ is the measurement received by the estimator and $\xi(k) \sim Ber(p)$ is a Bernoulli random variable that takes the value 1 with probability p and 0 with probability $1 - p$. That is, when $\xi(k)$ is 1, there is no packet dropout. If $\xi(k)$ is 0, however, the latest output is used at current time, in a recursive fashion.

Another way of describing multiple packet dropouts is by limiting the amount of consecutive dropouts (Shuli Sun et al., 2008), where the received measurements are

defined by

$$
\begin{aligned}
y(k) = &\xi(k)z(k) + (1 - \xi(k))\xi(k - 1)z(k - 1) + \ldots \\
&+ (1 - \xi(k))(1 - \xi(k - 1))\ldots(1 - \xi(k - N + 1))z(k - N), \quad N \geq 1.
\end{aligned}
\tag{3.8}
$$

Such a model dictates that the measurement used by the estimator will be only the most recent available, and the amount of missing observations is limited to N. This conclusion can be drawn by the fact that

$$
\xi(k) + (1 - \xi(k))\xi(k - 1) + \ldots + (1 - \xi(k))(1 - \xi(k - 1))\ldots(1 - \xi(k - N + 1)) = 1,
\tag{3.9}
$$

which means that if the Bernoulli RV realization for the most recent measurement available is 1, all past measurement terms will be multiplied by 0.

3.3.3 Uncertain Observation

For some applications, there is a chance that the observation signal sent to the estimator contains only noise. According to (Jaffer and Gupta, 1971), it happens as a consequence of two situations: the observation was taken, but it was lost during transmission, due to communication failures; or it was not transmitted at all, as it may happen for target tracking systems, for example, when the object being observed is not tracked at a sample time. A summary of causes and effects that time delay causes in an estimator are illustrated in Figure 3.5.

An observation model for a sampled-data system with uncertain observations can be described as

$$
y(k) = \xi(k)C(k)x(k) + D(k)v(k)
\tag{3.10}
$$

where $\xi(k) \sim Ber(p(k))$ is a Bernoulli random variables, taking values of 0 or 1, with probabilities $p(k)$ and $1 - p(k)$, respectively.

Unlike the packet dropout problem, when the missing data are associated with the total absence of signal, the issue of uncertain observation has to be dealt with differently. A common approach is to detect the existence of signal prior to the assimilation, using a likelihood ratio test. (Middleton and Esposito, 1968) proposes a joint approach to systematically detect and extract information from observation signals. If the estimator and detector are developed separately, the probability of false alarms is not used in
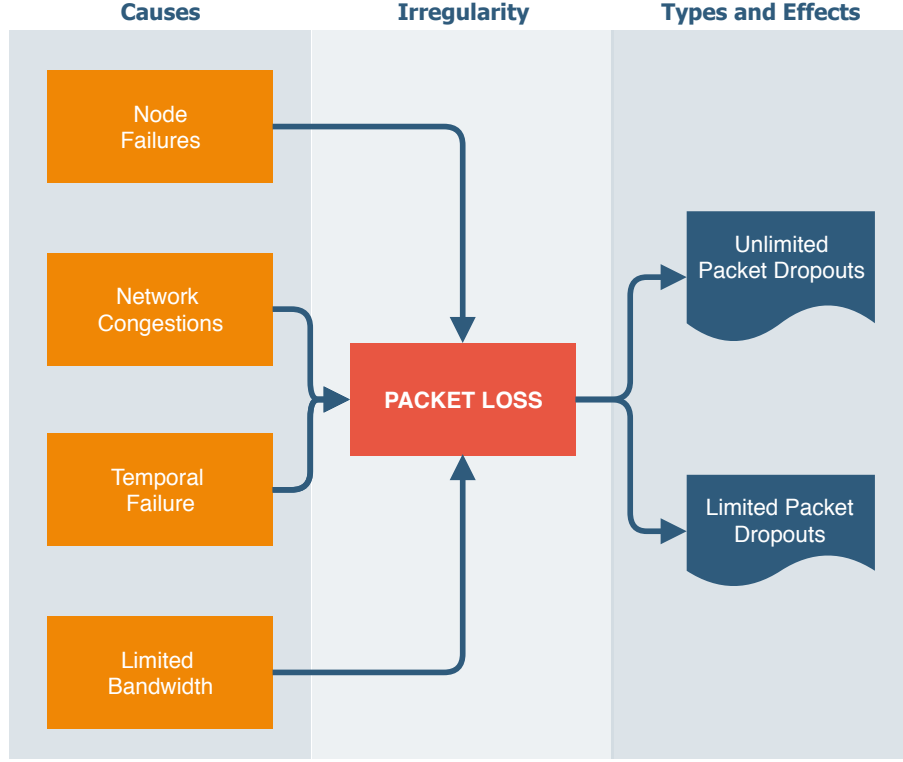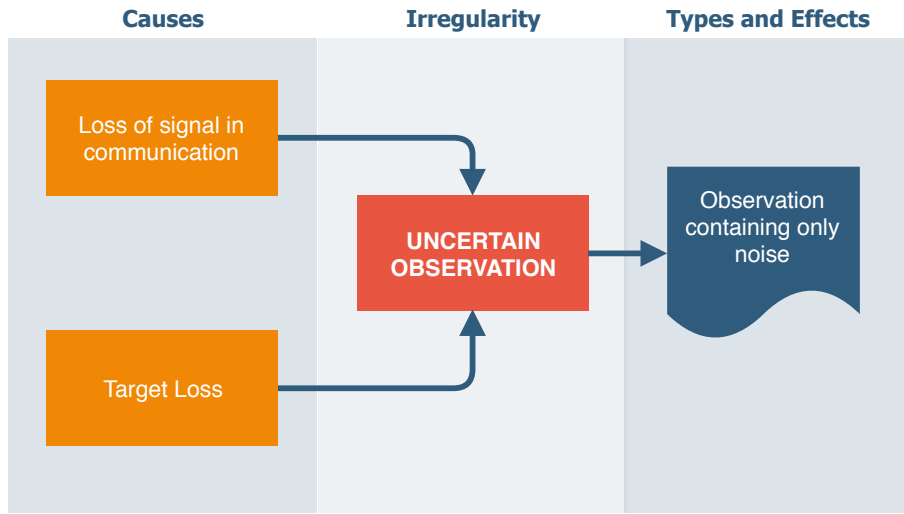
Figure 3.5: Uncertain observations diagram, showing the main causes (in orange) and effects (in dark blue) of time delay irregularity.

the estimator, making it sub-optimal. (Nahi, 1969) developed an optimal recursive estimator, that uses the information of the random variable γ in the algorithm, assuming its sequence is independent and identically distributed. (Hadidi and Schwartz, 1979) generalized the work of Nahi, for the case when the uncertainty of the signals presence is described by a Markovian sequence of binary random variables.

3.3.4   Aperiodic Sampling

All irregularities discussed so far may be present even in a periodic sampling scheme. However, for some applications, the sampling intervals are time-varying due to a variety of phenomena, causing the so-called aperiodic or asynchronous sampling. A summary of causes and types of aperiodic sampling is presented Figure 3.6.

Usually such irregularity happens in networked and embedded control systems, with unpredictable networked-induced issues, such as irregular faults on samplers, oscillated loads, intermittent saturation or even variations in system components or parameters (Shen et al., 2016). Some imperfections may cause what is known as sampling jitter noise, which leads to time intervals being almost uniform. Automotive applications, radar imaging or event controlled systems are a few examples. In such cases, jitter noise happens due to a sampling frequency similar to the clock frequency; to sampling requests delayed by the network; or to imperfect synchronization (Eng and Gustafsson, 2005). For networks with a large amount of unsynchronized sensors,
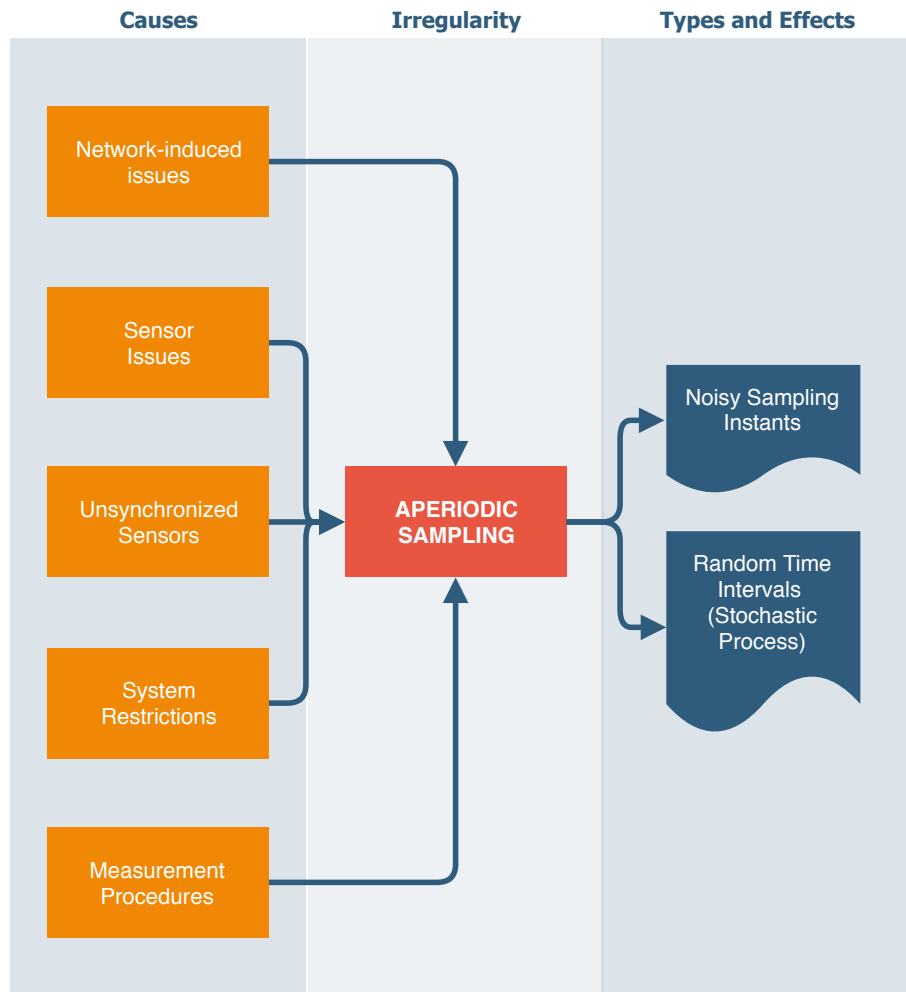
Figure 3.6: Aperiodic sampling diagram, showing the main causes (in orange) and types (in dark blue) of irregularity.

measurement arrival time instants are randomly spaced and can be modeled as stochastic processes (Micheli and Jordan, 2002). Sometimes, the system being observed has particularities that causes the aperiodic sampling. One example is seismology, where the spatial coordinates are irregularly sampled, because of natural obstacles (Marvasti, 2001). Other large scale systems, such as power grids, have sensors with huge geographical separations and different communication links to the estimation hub, causing multiple and random inter-observation intervals (Yan et al., 2017).

Whereas for most cases the deformities in sampling time intervals appear as unwanted effects, there are cases when the sampling rule is designed to work aperiodically. If there are limitations of communication resources (limited bandwidth or computation capacity, for instance) or a need for reduced energy consumption, time-driven

sampling might be neglected in favor of an event-based scheme. In such strategy, an event-triggering mechanism is responsible for determining the sampling instants, according to Figure 3.7. For time-driven schemes, a clock triggers the transmission instants, while event-driven sampling instants depends on the sensor output itself with an optional feedback loop from the estimator, to assess estimation performance. Therefore, the trigger mechanism design provides a trade-off between performance and resource consumption efficiency, attracting a lot of research interest (Liu et al., 2014). The most common strategy for event-driven state estimation is the send-on-delta (SOD) (Miskowicz, 2006), which triggers the transmission when the value of the measured state deviates from the previous assimilated observation by an interval $\pm\Delta$, with $\Delta > 0$. Other strategies were studied in (Zou et al., 2017). To avoid the risk of unexpected high amount of triggered measurements in a short period of time, which can lead to the dreaded Zeno behavior (Tabuada, 2007), lower-bounds can be defined both for the Δ-value or for some explicit minimum inter-event time.

(a)

(b)
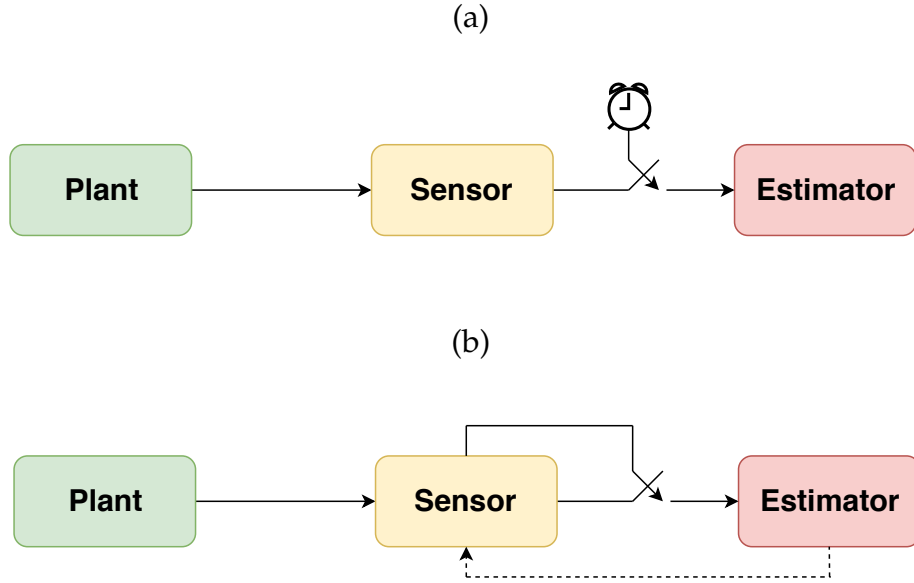
Figure 3.7: (a) Time-driven and (b) event-driven sampling schemes. The connection between sensor and estimator is triggered by different mechanisms.

The measurement model of a linear system with aperiodic sampling can be defined as

$$y(t_k) = H(t_k)x(t_k) + v(t_k) \tag{3.11}$$

where t_k is the random sampling time instant and the observation model matrix $H(t_k)$ is time-varying.

Generalizations of aperiodic sampling can be divided in two categories, based on how the estimator perceives the irregularity: as time noise added to a periodic pattern; or as a stochastic process, according to Figure 3.8.
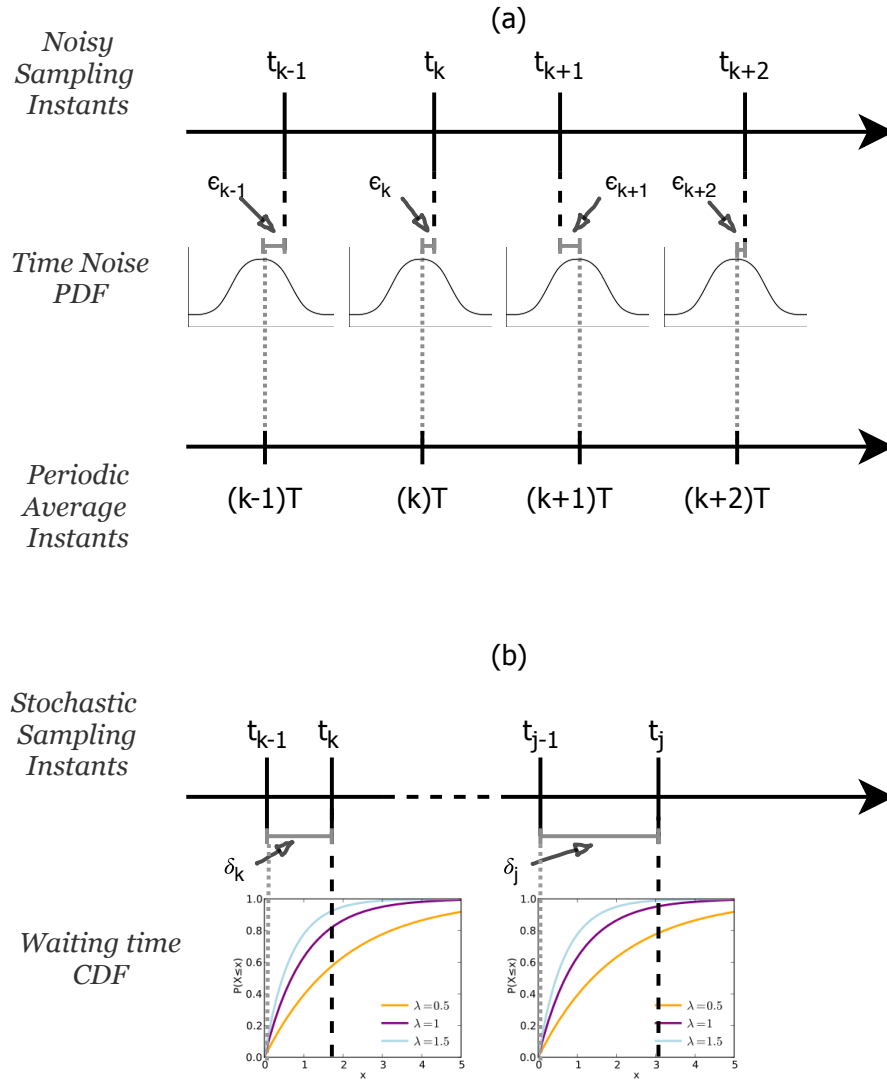
Figure 3.8: Two models of aperiodic sampling: (a) noisy sampling over periodic intervals, with a Gaussian random variable added to expected time instants kT; and (b) sampling instants modeled as a stochastic process, with time intervals characterized by an exponential random variable. For (b), the cumulative distribution functions are shown, for different λ parameter values.

For the first case, random time instants t_k and the random time intervals δ_k can be

defined as:

$$t_k \triangleq kT + \epsilon_k,$$
$$\delta_k \triangleq t_k - t_{k-1} \tag{3.12}$$

where t_k is the kth sampling instant, T is the periodic time interval and ϵ_k is the deviation from the expected value kT. Note that, if the sampling time instants are a sequence of i.i.d Gaussian random variables, with variance σ^2, that is $t_k \sim \mathcal{N}(kT, \sigma^2)$, $\forall k \sim \mathbb{N}$, then the time interval random variable is Gaussian, with expected value T and variance $2\sigma^2$, that is $\delta_k \sim \mathcal{N}(T, 2\sigma^2)$.

For the stochastic process generalization, sampling time instants t_k can be defined by the random time intervals δ_k, such as:

$$\delta_k \triangleq t_k - t_{k-1},$$
$$\delta_0 \triangleq t_1 \tag{3.13}$$

where random time intervals δ_k can be modeled, in the most flexible way, as a gamma probability density function, that is $\delta_k \sim \Gamma(\kappa, \theta)$. If the shape parameter κ is a positive integer, then it becomes an Erlang distribution, as set in (Kanchanaharuthai and Wong-saisuwan, 2002). For the most common case, where κ is held constant, the random time interval δ_k follows an exponential PDF and the time sequence t_k is represented by a Poisson stochastic process (Micheli and Jordan, 2002). In fact, Micheli and Jordan provided mathematical proof that for a network with N unsynchronized sensors with sampling period T, the waiting time between two arrivals tends, in distribution, to the exponential random variable, that is $\delta_k \sim \mathcal{E}(\lambda)$, where $\lambda = N/T$, as N tends to infinity. In such cases, the expected value of the RV d_k is $E[d_k] = 1/\lambda$ and its variance is $Var[d_k] = 1/\lambda^2$.

3.3.5 Multi-Rate Sampling

The last irregularity discussed is the multi-rate sampling. Generally, it refers to multiple sensors measuring variables from the same system at different sampling rates. Figure 3.9 shows a schematic of causes and types of the multi-rate sampling irregularity.

Many industrial processes need to control variables that can be measured by online
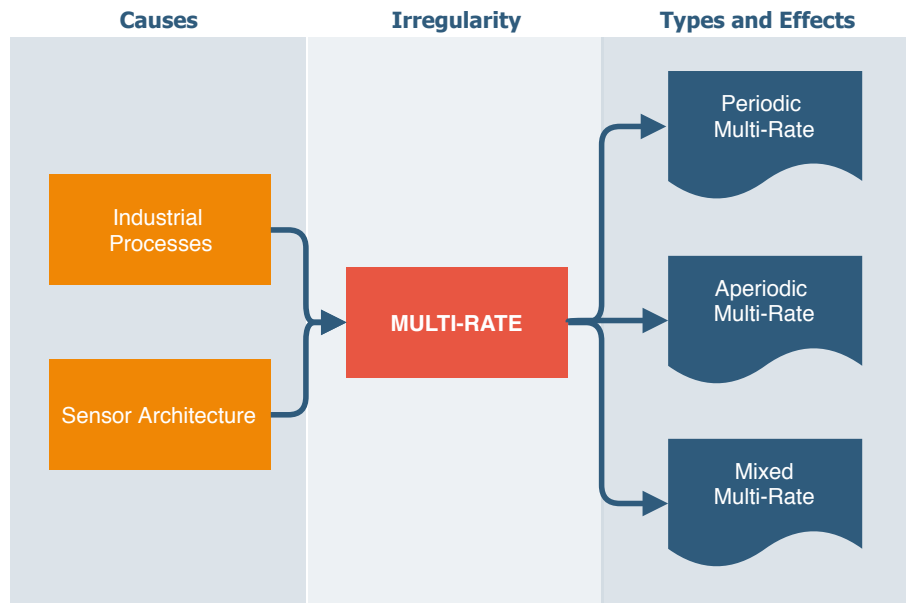
Figure 3.9: Multi-rate sampling diagram, showing the main causes (in orange) and types (in dark blue) of multi-rate sampling.

instruments that provide regular, fast rate and delay free information, but with low precision. Therefore, more accurate data are needed and usually available after slow, irregular and human-dependent laboratory analysis (Peñarrocha et al., 2012; Fatehi and Huang, 2017). The combination of both sources of measurements leads to a multi-rate sampling scenario.

A more common approach is the use of various sensors measuring the same physical information, to obtain better estimates, which has been drawing attention from real world applications, such as target tracking, robotics, surveillance and military. For such strategy, the sampling rates perceived by the estimator are often different from one another. The work of (Lin and Sun, 2016) and the references therein provide a wide coverage of scenarios derived from multi-sensor multi-rate systems.

Figure 3.10 illustrates the ways multi-rate sampling can be manifested in a system. The different rates from the various sensor devices can be periodic (a), aperiodic (b) or even a mixture of both, as it is the case for most industrial applications with laboratory analysis.

Aperiodic sampling rates can be described the same way as in Section 3.3.4, see (3.11). Periodic multi-rate measurements can be modeled as
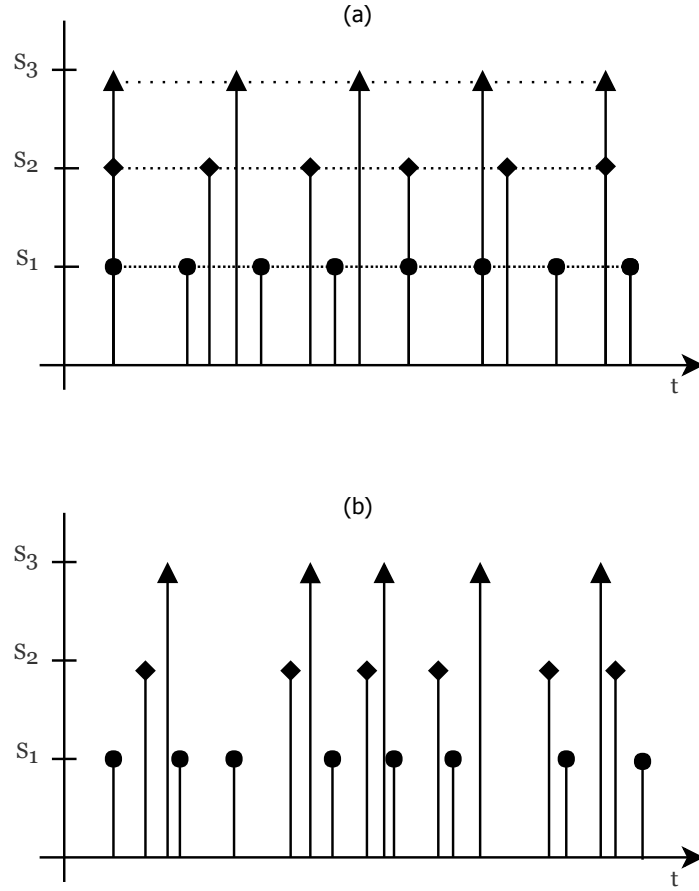
Figure 3.10: (a) Periodic and (b) aperiodic multi-rate sampling scheme. Labels S_1, S_2, S_3 refer to the sampling instants of three different sensors, from the highest sampling rate to the lowest sampling rate.

$$y_i(k_i T_i) = H_i(x(k_i T_i)) + v_i(k_i T_i) \tag{3.14}$$

where $y_i(k_i T_i)$ represents the k_i^{th} observation from sensor i with periodic sampling rate T_i and H_i is the discrete measurement model matrix related to that sensor.

3.4   Time Synchronization

There are techniques to handle all the irregularities discussed in this chapter and still achieve efficient state estimation performance. Provided that the estimator acknowledges the time that measurements were taken, there are many algorithms that provide interesting results. For some examples, we encourage the reader to check the

references from each subsection in this chapter.

In general, multi-sensor data fusion techniques for state estimation require that exact measurement time-stamps are available in order to assimilate data properly (Ping, 2003; Brahmi et al., 2013). However, that is not always the case and situations may arise in which time-stamps were not collected or their values are unreliable. Examples of the latter may occur when measurements are time stamped when they are received by the estimator instead of the moment it was taken, or when they are time stamped at local clocks without centralized synchronization (Julier and Uhlmann, 2005).

In many practical applications, if sampling irregularity cannot be accounted for accordingly, data are fused using the time of arrival as time-stamp (Huck et al., 2011), or irregularities such as OOSM are just disregarded completely (Kwok et al., 2004). The effects of such neglect has seldom been investigated, however. In one of the few studies, (Julier and Uhlmann, 2005) considered the state estimation problem for delayed but periodic observations with imprecise and unknown time-stamps, with the assumption that they could be statistically characterized. They proposed an implementation of the covariance union (CU) technique, using the time-stamp statistics in the filter update step. First, the method calculates the updates considering both the maximum and minimum expected delayed and then merged both results with a convex combination designed to minimize the state covariance matrix. CU algorithm was tested for the problem of estimating the states of a linear system, considering a random time delay, uniformly distributed between 2 and 10 time steps. Estimation performance was tested against four other methods: *known delay*, that considers the time delay to be known perfectly, as a baseline for comparison; *mean delay*, that assumes the delay to be always an average of 6 time steps; maximum likelihood, that calculates the likelihood of each possible time step and selects the highest; and Probabilistic Data Association Filter (PDAF) (Bar-shalom et al., 2009), that also calculates the likelihood of each step and averages the results using the likelihood as weight. They used the normalized state estimation error (NEES) test for consistency assessment, which we explain later in Section 4.4, for a linear system with two states. Therefore, the expected value of NEES for a consistent filter should be 2. The results are replicated in Table 3.1 for all algorithms.

As expected, the method with known delay was the most consistent, that is with E [NEES] results closer to 2. CU and PDAF were both consistent, with slightly overestimated of the state error. They argued that, although PDAF obtained better consistency

Table 3.1: Comparison of NEES consistency test results for fusion of time delayed measurements with uncertain time-stamps for a linear system with two states

Method	E[NEES]
Known delay	1.9992
Mean delay	37.6949
Max likelihood	54.9323
Covariance union	1.3172
PDAF	1.8749

results, it has significantly higher computational costs. Nevertheless, the results of the mean delay and maximum likelihood are highly inconsistent, showing that the neglect or oversimplification of the imprecision in time-stamp leads to considerable performance degradation.

Another approach to overcome unknown time-stamps on the presence of sampling irregularities was studied by (Kwok et al., 2004). In the presence of OOSM, when sensor information is sampled at much faster rates than filter update rates, the real-time particle filter (RTPF) proposed by them, makes efficient use of all sensor information, instead of discarding sensor readings. That is achieved by dividing the received measurements among sample sets and then representing the states as a mixture of those sets.

Alternatively, in order to avoid performance degradation, one can make use of time synchronization schemes, widely used in communication networks, to ensure global time stamps. Wireless sensor networks (WSN) are particularly dependent on such techniques, due to limited computation, energy and communication resources of the sensing devices used. The work of (Sivrikaya and Yener, 2004) provide detailed reviews of the time synchronization problem in sensor networks. They explain the problem through computer clock mechanism.

With the aid of a hardware oscillator, local clocks from a sensing device node i approximates real time t as $C_i(t)$ by

$$C_i(t) = a_i t + b_i \tag{3.15}$$

where $a_i(t)$ is the clock *drift*, that is the clock frequency, and $b_i(t)$ represents an *offset* value, or the difference from real value t.

Clock approximations from two nodes in a network are compared by

$$C_1(t) = a_{12}C_2(t) + b_{12} \tag{3.16}$$

where a_{12} is the relative *drift* and b_{12} is the relative offset between nodes.

Under equations (3.15) and (3.16), the synchronization problem becomes the equalization of the computer clocks from all different n devices, in its most strict form. Thus, synchronization strategies can either match all clock frequencies and offsets once or perform repeated offset corrections over time. There are more relaxed versions of synchronizations, such as the one proposed by Römer (2003), that aims at maintaining only the order of events.

Probably the most popular time synchronization method is the one being used in the internet environment for years, the Network Time Protocol (NTP), designed by (Mills, 1991). For most control and WNS applications, however, it is not suitable, due to very different requirements, such as energy consumption, precision and scalability (Ganeriwal et al., 2003). An easy solution would be to equip all sensing devices in the network with a global positioning system (GPS) for a global time synchronization, but such solution is very expensive, not energy efficient and its signal might not work properly in every environment.

Therefore, many alternative methods have been proposed, and the work of (Kaur and Abhilasha, 2015) updates the studies from (Sivrikaya and Yener, 2004), with an exploration of the most recent synchronization protocols for sensor networks, that is: reference broadcast synchronization (RBS) (Elson et al., 2002); timing-sync protocol for sensor networks (TPSN) (Ganeriwal et al., 2003); delay measurement time synchronization (DMTS) (Ping, 2003); lightweight tree-based synchronization (LTS) (van Greunen and Rabaey, 2003); tiny-sync mini-sync (Sichitiu and Veerarittiphan, 2003); flooding time synchronization protocol (FTSP) (Maróti et al., 2004); lightweight and energy efficient time synchronization (LEETS) (Mingxia Xu et al., 2005); time diffusion protocol (TDP) (Su and Akyildiz, 2005); and time synchronization based on spanning tree (TSST) (He, 2008). A comparison adapted from Kaur ad Abhilasha is presented in Table 3.2.

Table 3.2: Comparison of time synchronization methods. Parameters are *Precision, Energy Efficiency (E. E.) and Complexity (Comp.).* Adapted from (Kaur and Abhilasha, 2015)

Protocol	Advantages	Limitations	Parameters
RBS	Eliminates random delays on the sender side	High amount of message exchanges and low transmission range	29.1 μs High *E. E.* High *Comp.*
TPSN	Eliminates the access, byte alignment and propagation times	Does not estimate the clock drift; does not handle dynamic topology changes and demands high communication load	16.9 μs High *E. E.* Low *Comp.*
DMTS	Reduces the number of message exchanges	Restricted to low resolution and low frequency external clocks	32 μs V. High *E. E.* Low *Comp.*
LTS	Robust and works well in the presence of dynamic links and fading.	The accuracy of synchronization decreases linearly in the depth of the synchronization tree	Unknown Low *E. E.* Low *Comp.*
Tiny-Sync Mini-Sync	Tolerant to message losses and adequate for networks with limited bandwidth and computational power	Unsuited for mobile sensor networks, high convergence time, not scalable and little robustness	945 μs High *E. E.* Low *Comp.*
FTSP	Robust, handles dynamic topology changes well and eliminates maximum delay components	Does not eliminate propagation delay and is not scalable	1.48 μs High *E. E.* High *Comp.*
LEETS	Low power consumption and low amount of message exchanges	Requires a GPS receiver in the root node	30 μs High *E. E.* Low *Comp.*
TDP	Tolerant to message losses, high mobility and performs well even without external servers	Very high convergence time	100 μs High *E. E.* High *Comp.*
TSST	Low synchronization error	Not scalable	Unknown Low *E. E.* Low *Comp.*

3.5   Chapter Summary and Final Remarks

In this chapter, the main sampling irregularities are reviewed: time delay, packet loss, uncertain observation, aperiodic and multi-rate. Diagrams describing their causes, types and effects are shown for each of them. We also describe the necessary modifications to the observation models of state estimation algorithms. Most of the methods proposed in the literature to handle sampling irregularities rely on the correct time stamping of observations. Thus, time synchronization in sensor networks becomes crucial and its further explored. The most recent protocols developed to ensure a global time scale from sensing devices in large sensor networks are shown and compared.

However, the use of any time synchronization method will require computational, energy and resource consumption to some extent, apart from complex algorithms implementations. For sensor fusion performance applications in state estimation of sampled-data systems with irregular sampling, the investment might not be worth it. Thus, the next chapters try to shed some light in the evaluation of performance degradation in state estimation in the presence of irregular sampling, if time-stamps are not available.

State Estimation for Sampled-Data Systems

In this chapter, the field of sensor fusion under irregular sampling is resumed to the problem of estimating the states of a system with a known process model, that is observed by aperiodically sampled measurements. We begin by describing the discrete-time representation for sampled-data systems. Then we review the adopted state estimation algorithm, that is the Kalman Filter, which is the most common approach to probabilistic data fusion. The nonlinear extension based on the unsigned transform is also explored. In Section 4.3 we describe the particularities of the filtering algorithms for when the correct time-stamp is available to the estimator and when it is not. We end with a description of the performance criteria used for assessment of the results, designed to quantify estimation accuracy and consistency.

4.1 Discrete-Time Representation of Sampled-Data Systems

State estimation for nonlinear sampled-data systems can be performed using the continuous-time process model and the discrete-time observation model. The state-estimate propagation is performed using numerical integration over the time interval defined by the arrivals of two consecutive measurements. The work of (Teixeira et al., 2008) compares two of those methods, the sampled-data extended Kalman Filter (SDEKF) and the sampled-data unscented Kalman Filter (SDUKF) to estimate the trajectory of a satellite.

To avoid computational complexity of the numerical integrations, discretization of the process model is widely used in state estimation applications (Aguirre et al., 2005), which motivates the use of discrete-time methods in this study. Thus the system

described in Section 1.2 needs to be formulated as a discrete-time model. First, we reproduce the stochastic nonlinear sampled system models as follows

$$\dot{x}(t) = f(x(t),u(t),w(t),t), \tag{4.1}$$

$$y(t_k) = g(x(t_k),v(t_k),t_k), \tag{4.2}$$

where $f \colon \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}^+ \to \mathbb{R}^n$ and $g \colon \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^+ \to \mathbb{R}^m$ are, respectively, the process and observation models, $k \in \mathbb{N}$ is the discrete-time index, t_k represents the irregular time instants, $x(t) \in \mathbb{R}^n$ the state vector, $u(t) \in \mathbb{R}^p$ is the input vector and $w(t) \in \mathbb{R}^q$ is the process noise vector, all at continuous-time t, $y(t_k) \in \mathbb{R}^m$ represents the irregularly sampled measurement sequence and $v(t_k) \in \mathbb{R}^r$ the corresponding measurement noise vector. Initial condition x_0 is assumed to be known.

The continuous-time system dynamics defined by (4.1) is discretized replacing the differentiation by differencing, via numerical methods. For this study we use the fourth order Runge-Kutta method (Süli and Mayers, 2003), given by

$$x(t_{k+1}) = x(t_k) + \frac{1}{6}\left(k_1 + 2k_2 + 3k_3 + k_4\right), \tag{4.3}$$

$$t_{k+1} = t_k + h_k, \tag{4.4}$$

where $h_k \in \mathbb{R}$ is the variable step size, defined by random time intervals, and

$$k_1 = h_k f(t_k, x(t_k)), \tag{4.5}$$

$$k_2 = h_k f(t_k + \frac{h_k}{2}, x(t_k) + \frac{k_1}{2}), \tag{4.6}$$

$$k_3 = h_k f(t_k + \frac{h_k}{2}, x(t_k) + \frac{k_2}{2}), \tag{4.7}$$

$$k_4 = h_k f(t_k + h_k, x(t_k) + k_3), \tag{4.8}$$

where $f(x(t),u(t),w(t),t)$ was reduce to $f(x(t),t)$ for simplicity, since the numerical approximation depends only on the vector state itself and the time intervals. Input $u(t)$ is considered to be constant during the time step and $w(t)$ represents unknown noise. Therefore, the discrete-time representation for the nonlinear sampled system becomes

$$x(t_{k+1}) = f_{\mathrm{d}}(x(t_k), u(t_k), w(t_k), t_k), \tag{4.9}$$

$$y(t_k) = g(x(t_k), v(t_k), t_k), \tag{4.10}$$

where $f_{\mathrm{d}}(\cdot)$ is given by (4.3)-(4.8). Note that even if the nonlinear dynamics given by (4.1) is time-invariant, that is if $f(\cdot)$ does not depend directly on t, its discrete counterpart will be time-varying if the system is irregularly sampled, since there is a time-step dependent factor h_k outside $f(\cdot)$ for each k_n, $n = [1,2,3,4]$, in (4.5)-(4.8). Discrete time instants t_k are determined by random time intervals h_k, that is $t_{k+1} = t_k + h_k$, $\in \mathbb{R}$ as in (4.4) and $k \in \mathbb{N}$ is the discrete-time index.

For the linear time-invariant case (LTI), dynamic model of the sampled system is given by

$$\dot{x}(t) = Ax(t) + Bu(t) + Gw(t), \tag{4.11}$$

$$y(t_k) = Cx(t_k) + v(t_k), \tag{4.12}$$

where matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times p}$, $G \in \mathbb{R}^{n \times q}$ and $C \in \mathbb{R}^{m \times n}$ are known matrices, $\rho(w(t)) = \mathcal{N}(0,Q)$ and $\rho(v(t_k)) = \mathcal{N}(0,R)$ are process and measurement noises, respectively, with covariance matrices $Q \in \mathbb{R}^{q \times q}$ and $R \in \mathbb{R}^{r \times r}$. If input $u(t)$ is digitally generated at the same rate as observations are taken, and followed by a digital-to-analog converter, then it is piecewise constant. In other words $u(t) = u(t_k)$, for $t_k \leq t < t_{k+1}$. In such cases, there is an exact solution to the discretization problem of (4.11), given by (Chen, 1999)

$$x(t_{k+1}) = A_{\mathrm{d}}x(t_k) + B_{\mathrm{d}}u(t_k) + w_{\mathrm{d}}(t_k), \tag{4.13}$$

where $t_{k+1} = t_k + h_k$, $\in \mathbb{R}$ are discrete time instants as in (4.4), separated by random time intervals $h_k \in \mathbb{R}$ and $k \in \mathbb{N}$ is the discrete-time index. The matrices from the discrete-time representation and the input noise $w_d(t_k)$ are given by

$$A_{\mathrm{d}} = e^{Ah_k}, \tag{4.14}$$

$$B_{\mathrm{d}} = \left(\int_{\tau=0}^{h_k} e^{A\tau} \delta\tau \right) B, \tag{4.15}$$

$$w_{\mathrm{d}}(t_k) = \int_{\tau=0}^{h_k} e^{A(t_{k+1}-\tau)} G dw(\tau) d\tau,. \tag{4.16}$$

Computation of A_d, given by the matrix exponential (4.14), can be solved by expansion in Taylor series

$$e^{Ah_k} = I + h_k A + \frac{h_k^2 A^2}{2!} + \dots + \frac{h_k^n A^n}{n!} + \dots = \sum_{i=0}^{\infty} \frac{h_k^i A^i}{i!}. \tag{4.17}$$

By truncating (4.17) at some point, A_d can be approximated by matrix multiplications and additions. Taylor series can also be used to compute B_d. The integral factor of (4.15) can be expanded as

$$\int_{\tau=0}^{h_k} e^{A\tau} \delta\tau = \int_{\tau=0}^{h_k} \left(I + \tau A + \frac{\tau^2 A^2}{2!} + \dots \right) d\tau,$$
$$= h_k I + \frac{h_k^2 A}{2!} + \frac{h_k^3 A^2}{3!} + \dots, \tag{4.18}$$

If A is nonsingular, the series can be written as

$$\int_{\tau=0}^{h_k} e^{A\tau} \delta\tau = A^{-1} \left(A h_k I + \frac{h_k^2 A^2}{2!} + \frac{h_k^3 A^3}{3!} + \dots + I - I \right),$$
$$= A^{-1} \left(e^{A_k} - I \right). \tag{4.19}$$

Thus uniting (4.15) and (4.19), B_d becomes

$$B_d = A^{-1} (A_d - I) B). \tag{4.20}$$

If input noise $w(t)$ from (4.11) is zero-mean white Gaussian, then $w_d(t_k)$ given by (4.16) will also be zero-mean white Gaussian (Jazwinski, 1970), with covariance Q_k given by

$$Q_k = \int_{\tau=0}^{h_k} e^{A\tau} GQG^T e^{A^T\tau} d\tau, \tag{4.21}$$

For state estimation purpose, we are interested in the calculation of (4.21) at every time instant t_k, $k \in \mathbb{N}$, given that exact information about timestamp is available. Usually, for regularly sampled systems, discrete noise covariance is either modeled or tuned directly (Wahlström et al., 2014). However, if the system is sampled aperiodically, the dependency between covariance Q_k and time interval h_k must be taken into account. The calculation of (4.21) is nontrivial and there are numerical methods available for computing the matrix exponential integral, at the expense of high computational costs (Micheli and Jordan, 2002). In this study we simplify the problem, adopting a common practice in state estimation applications, that employs approximate discretization (Axelsson and Gustafsson, 2015). Continuous-time noise is assumed to be piecewise constant in the interval h_k, that is $w(t) \approx w(t_k)$, for $t_k \leq t < t_{k+1}$. Then, (4.11) is discretized by

$$x(t_{k+1}) = A_{\mathrm{d}} x(t_k) + B_{\mathrm{d}} u(t_k) + G_{\mathrm{d}} w_{\mathrm{d}}(t_k), \tag{4.22}$$

where A_{d} and B_{d} are given by (4.14) and (4.15) respectively, G_{d} is calculated the same way as B_{d}, that is $G_{\mathrm{d}} = \left(\int_{\tau=0}^{h_k} e^{A\tau} \delta\tau \right) G$ and $Q_k = h_k Q$.

Note that, once again, even for the time-invariant case, the matrices from the discrete-time system depend on time intervals, thus will be time-variant if sampling is aperiodic.

Usually, the covariance matrices Q and R of the noise vectors, that represent the noise level, are better expressed by a signal-to-noise ratio (SNR) quantity, which relates more directly to how contaminated is a signal. It is basically a relation between the power level of the signal and of the noise, that is

$$SNR = \frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}, \tag{4.23}$$

where P is the average power of a signal, which is directly proportional to its variance. Therefore it can also be expressed by

$$SNR = \frac{\sigma_{\mathrm{signal}}^2}{\sigma_{\mathrm{noise}}^2}. \tag{4.24}$$

In Chapter 5 we use SNR definition to express the noise level present in signals, using the logarithmic decibel (dB) scale, given by

$$SNR_{\text{dB}} = 10 \log_{10} \frac{P_{\text{signal}}}{P_{\text{noise}}}, \tag{4.25}$$

In the next section we introduce the estimation problem and the Bayesian solution, considering the discrete-time models of state space representations. For simplicity, we replace (t_k) argument for a subscript k, to represent discrete instants of time.

4.2   Bayesian Estimation

The Bayesian approach to state estimation can be interpreted as a data fusion algorithm in which the inferred knowledge about the system's states is updated as new information arrives, using not just the new data, but also the prior information. For that, both the states and the arrived information are modeled as random variables (RV), hence its classification falls under the probabilistic fusion framework.

The goal of the estimator is to statistically and recursively infer the values of the system's states, that is the random vector $x_k \in \mathbb{R}^n$, from noisy data, that is the observation vector sequence $y_1, \ldots, y_k \in \mathbb{R}^m$ and input vector sequence $u_1, \ldots, u_k \in \mathbb{R}^p$. The corresponding conditional PDF, $\rho(x_k|(y_1, \ldots, y_k))$, is called the *posterior* PDF, describing the statistics of the random vector $x_k \in \mathbb{R}^n$, after the present and past experimental observations $\rho(x_k|(y_1, \ldots, y_k))$ have been assimilated and the process model has been taken into account. Thus we can find the estimation of x_k using methods such as the *maximum a posteriori* (MAP) or *minimum mean square error* (MMSE), given by (Bar-Shalom et al., 2001)

$$\hat{x}_k^{MAP} \triangleq \arg \max_{x_k} \hat{\rho}(x_k|(y_1, \ldots, y_k)), \tag{4.26}$$

$$\hat{x}_k^{MMSE} \triangleq \arg \min_{x_k} E[(\hat{x}_k - x_k)^T(\hat{x}_k - x_k)|(y_1, \ldots, y_k)], \tag{4.27}$$

where $\hat{x}_k$ is the estimated value of $x_k \in \mathbb{R}^n$, $\rho(x_k|(y_1, \ldots, y_k))$ is the estimated posterior PDF of x_k given the observation sequence, and $E[(\hat{x}_k - x_k)^T(\hat{x}_k - x_k)|(y_1, \ldots, y_k)]$ is the variance of the random vector x_k, given the observation sequence. MAP can be interpreted as

the bayesian approach to *maximum likelihood* (ML) estimation, whereas MMSE is the counterpart of the *least squares* (LS) estimator (Bar-Shalom et al., 2001).

Finding the *posterior* $\rho(x_k|(y_1,\ldots,y_k))$ defines the complete state estimation problem, whereas the estimates $\hat{x}_k^{MAP} \in \mathbb{R}^n$ and $\hat{x}_k^{MMSE} \in \mathbb{R}^n$ are the optimal state estimates, under their optimality criteria. A recursive Bayesian solution to the state estimation problem, that is finding the *posterior* PDF, considering that the system evolves according a Markov process is presented in Proposition 4.2.1. But first, we need to define two lemmas.

Lemma 4.2.1. For a Markov system with initial state $x_0 \sim \rho(x_0)$, the transition PDF of the future state x_{k+1} given the present state x_k is independent of past states, that is

$$\rho(x_{k+1}|(x_0,\ldots,x_k)) = \rho(x_{k+1}|x_k), \tag{4.28}$$

and observation vector y_k is independent of past observations and past states, given the present state x_k, that is

$$\rho(y_k|(x_0,\ldots,x_k,y_0,\ldots,y_{k-1}) = \rho(y_k|x_k). \tag{4.29}$$

$\square$

Lemma 4.2.2. For a Markov system, we can find the transition PDF from step n to step s as a function of the transition densities between them and an intermediate step r, as long as $n > r > s$, by the Chapman-Kolmogorov equation (Papoulis, 1984)

$$\rho(x_n|x_s) = \int_{-\infty}^{\infty} \rho(x_n|x_r)\rho(x_r|x_s)dx_r. \tag{4.30}$$

$\square$

Note that the systems described in Section 4.1 are Markovian, since the transition or process models given by (4.9) and (4.12) and the observation models given by (4.10) and (4.12) follow the conditions (4.28) and (4.29), respectively.

Proposition 4.2.1. The posterior PDF of system states $x_k \in \mathbb{R}^n$ conditioned on the observation vector sequence $y_1,\ldots,y_k \in \mathbb{R}^m$ is recursively given by

$$\rho(x_k|(y_1,\ldots,y_{k-1})) = \int_{\mathbb{R}^n} \rho(x_k|x_{k-1})\rho(x_{k-1}|y_1,\ldots,y_{k-1})dx_{k-1}, \tag{4.31}$$

$$\rho(x_k|(y_1,\ldots,y_k)) = \frac{\rho(y_k|x_k)\rho(x_k|(y_1,\ldots,y_{k-1}))}{\rho(y_k|(y_1,\ldots,y_{k-1}))} \tag{4.32}$$

where $k \in \mathbb{N}$, $\rho(y_k|x_k)$ is the *likelihood PDF*, $\rho(x_k|(y_1,\ldots,y_{k-1})$ is the *prior PDF*, defined before the latest measurement, $\rho(x_k|x_{k-1})$ is the *transition PDF* that models the evolution of x_k and $\rho(y_k|(y_1,\ldots,y_{k-1}))$ is the *evidence*, also referred to as normalizing factor, or marginalization.

<div align="right">□</div>

Proof. The posterior PDF can be computed by the *Bayes' rule* (Stone, 2013)

$$\rho(x_k|(y_1,\ldots,y_k)) = \frac{\rho((y_1,\ldots,y_k)|x_k)\rho(x_k)}{\rho(y_1,\ldots,y_k)}. \tag{4.33}$$

Using the definition of the conditional probability, given by (Papoulis, 1984)

$$\rho((a_1,\ldots,a_k)|(a_{k+1},\ldots,a_n)) = \frac{\rho(a_1,\ldots,a_k,\ldots,a_n)}{\rho(a_{k+1},\ldots,a_n)}, \tag{4.34}$$

and the chain rule of probability, that is (Papoulis, 1984)

$$\rho(a_1,\ldots,a_n) = \rho(a_n|a_1,\ldots a_{n-1})\rho(a_1,\ldots a_{n-1}), \tag{4.35}$$

it is possible to rewrite (4.33) as

$$\rho(x_k|(y_1,\ldots,y_k)) = \frac{\rho(y_k|(y_1,\ldots,y_{k-1},x_k))\rho(y_1,\ldots,y_{k-1}|x_k)\rho(x_k)}{\rho(y_k|(y_1,\ldots,y_{k-1}))\rho(y_1,\ldots,y_{k-1})}. \tag{4.36}$$

From Lemma 4.2.1, given the current state x_k, the present y_k is independent of past observations, thus the first term of the dividend becomes $\rho(y_k|x_k)$. Additionally, Bayes' rule in the second term yields

$$\rho((y_1,\ldots,y_{k-1})|x_k)) = \frac{\rho(x_k|(y_1,\ldots,y_{k-1}))\rho((y_1,\ldots,y_{k-1}))}{\rho(x_k)}. \tag{4.37}$$

Finally, by combining all together, we have

$$\rho(x_k|(y_1,\ldots,y_k)) = \frac{\rho(y_k|x_k)\rho(x_k|(y_1,\ldots,y_{k-1}))\rho((y_1,\ldots,y_{k-1}))\rho(x_k)}{\rho(y_k|(y_1,\ldots,y_{k-1}))\rho(y_1,\ldots,y_{k-1})\rho(x_k)}, \tag{4.38}$$

which, after canceling the equal terms (in gray), proves (4.32).

To prove (4.31), we introduce a predicted state x_{k+1} in the posterior PDF, that is $\rho(x_{k+1},x_k|(y_1,\ldots,y_k))$ (Bergman, 1999). Rewriting the new conditional PDF with the aid of (4.34), (4.35) and Lemma 4.2.1, we have

$$\rho(x_{k+1},x_k|(y_1,\ldots,y_k)) = \rho(x_{k+1}|(x_k,y_1,\ldots,y_k))\rho(x_k|(y_1,\ldots,y_k))$$
$$= \rho(x_{k+1}|(x_k)\rho(x_k|(y_1,\ldots,y_k)). \tag{4.39}$$

The integration of both sides of (4.39) with respect to x_k yields (4.31), which is also a Chapman-Kolmogorov equation. This last piece of proof is also an application of Lemma 4.2.2.

$$\square$$

The evidence term in the denominator of (4.32) is commonly presented as a constant (Särkkä, 2013), since it does not depend on the state vector x_k, and it can be computed by the Chapman-Kolmogorov equation from Lemma 4.2.2

$$\rho(y_k|(y_1,\ldots,y_{k-1})) = \int_{\mathbb{R}^n} \rho(y_k|x_k)\rho(x_k|(y_1,\ldots,y_{k-1}))dx_k, \tag{4.40}$$
$$= z_k.$$

The algorithm is initialized by a known prior $\rho(x_0)$ and recursion is achieved by introducing the PDF calculated in the *forecast* step, given by (4.31), in the *data assimilation* step, given by (4.32). Thus, (4.32) can be interpreted as the fusion of the *prior* PDF or knowledge of the state with the *likelihood* PDF or evidence. For a better understanding, we can illustrate this process for a one-dimensional Gaussian case (Faragher, 2012), for which the PDFs are given by

$$\rho(x_k|(y_1,\dots,y_{k-1})) = \mathcal{N}(x_k|\hat{x}_{\mathrm{p}},\hat{\sigma}_{\mathrm{p}})$$

$$= \frac{1}{\sqrt{2\pi}\hat{\sigma}_{\mathrm{p}}} \exp\left(-\frac{(x_k - \hat{x}_{\mathrm{p}})^2}{2\hat{\sigma}_{\mathrm{p}}^2}\right), \tag{4.41}$$

$$\rho(y_k|x_k) = \mathcal{N}(y_k|x_k,\hat{\sigma}_{\mathrm{e}})$$

$$= \frac{1}{\sqrt{2\pi}\hat{\sigma}_{\mathrm{e}}} \exp\left(-\frac{(y_k - x_k)^2}{2\hat{\sigma}_{\mathrm{e}}^2}\right), \tag{4.42}$$

where the subscript p stands for *prior* and the subscript e stands for *evidence*. The *prior* PDF defined in (4.41) is obtained by forecasting with the aid of a process model, and the *likelihood* PDF given by (4.42) is obtained by a linear observation model, say $y_k = x_k + w_k$, with $w_k \sim \mathcal{N}(0,\sigma_{\mathrm{e}})$. Under the MAP estimation method, we can combine (4.26) and (4.32), discarding the normalization factor, which is independent of x_k, yielding

$$\hat{x}_k^{MAP} = \arg\max_{x_k} \rho(y_k|x_k)\rho(x_k|(y_1,\dots,y_{k-1}))$$

$$= \arg\max_{x_k} \mathcal{N}(y_k|x_k,\hat{\sigma}_{\mathrm{e}})\mathcal{N}(x_k|\hat{x}_{\mathrm{p}},\hat{\sigma}_{\mathrm{p}}), \tag{4.43}$$

and from the multiplication of (4.41) and (4.42) we have that

$$\mathcal{N}(y_k|x_k,\hat{\sigma}_{\mathrm{e}})\mathcal{N}(x_k|\hat{x}_{\mathrm{p}},\hat{\sigma}_{\mathrm{p}}) = \frac{1}{\sqrt{2\pi}\hat{\sigma}_{\mathrm{p}}} \exp\left(-\frac{(x_k - \hat{x}_{\mathrm{p}})^2}{2\hat{\sigma}_{\mathrm{p}}^2}\right) \times \frac{1}{\sqrt{2\pi}\hat{\sigma}_{\mathrm{e}}} \exp\left(-\frac{(y_k - x_k)^2}{2\hat{\sigma}_{\mathrm{e}}^2}\right)$$

$$= \frac{1}{2\pi\hat{\sigma}_{\mathrm{p}}\hat{\sigma}_{\mathrm{e}}} \exp\left[-\left(\frac{(y_k - x_k)^2}{2\hat{\sigma}_{\mathrm{e}}^2} + \frac{(x_k - \hat{x}_{\mathrm{p}})^2}{2\hat{\sigma}_{\mathrm{p}}^2}\right)\right], \tag{4.44}$$

whose quadratic terms can be expanded and, apart from a constant term that will be handled by the normalization factor discarded in 4.43, is rearranged into

$$\mathcal{N}(y_k|x_k,\hat{\sigma}_{\mathrm{e}})\mathcal{N}(x_k|\hat{x}_{\mathrm{p}},\hat{\sigma}_{\mathrm{p}}) = \mathcal{N}(x_k|\mu_{\mathrm{fused}}(y_k),\hat{\sigma}_{\mathrm{fused}})$$

$$= \frac{1}{\sqrt{2\pi}\hat{\sigma}_{\mathrm{fused}}} \exp\left(-\frac{(x_k - \mu_{\mathrm{fused}}(y_k))^2}{2\hat{\sigma}_{\mathrm{fused}}^2}\right), \tag{4.45}$$

where

$$\mu_{\text{fused}}(y_k) \triangleq \hat{x}_{\text{posterior}}(y_k) = \frac{\hat{\sigma}_{\text{e}}^2}{\hat{\sigma}_{\text{p}}^2 + \hat{\sigma}_{\text{e}}^2}\hat{x}_{\text{p}} + \frac{\hat{\sigma}_{\text{p}}^2}{\hat{\sigma}_{\text{e}}^2 + \hat{\sigma}_{\text{p}}^2}y_k$$

$$= \hat{x}_{\text{p}} + \frac{\hat{\sigma}_{\text{p}}^2}{\hat{\sigma}_{\text{e}}^2 + \hat{\sigma}_{\text{p}}^2}(y_k - \hat{x}_{\text{p}}), \tag{4.46}$$

and

$$\hat{\sigma}_{\text{fused}} \triangleq \hat{\sigma}_{\text{posterior}} = \frac{\hat{\sigma}_{\text{p}}^2\hat{\sigma}_{\text{e}}^2}{\hat{\sigma}_{\text{p}}^2 + \hat{\sigma}_{\text{e}}^2}$$

$$= \hat{\sigma}_{\text{p}}^2 - \frac{\hat{\sigma}_{\text{p}}^2}{\hat{\sigma}_{\text{e}}^2 + \hat{\sigma}_{\text{p}}^2}\hat{\sigma}_{\text{p}}^2. \tag{4.47}$$

The derivation from (4.46) and (4.47) comes from the multiplication of two Gaussian PDFs. Combining the results of (4.46) and (4.47) with (4.43), we have that

$$\hat{x}_k^{MAP} = \arg\max_{x_k} \mathcal{N}(x_k|\hat{x}_{\text{posterior}}(y_k), \hat{\sigma}_{\text{posterior}})$$

$$= \hat{x}_{\text{posterior}}(y_k). \tag{4.48}$$

In other words, the MAP estimate will be the mean of the *posterior* Gaussian, with its new parameters determined by a weighted combination of the variances of both *prior* and *likelihood* PDFs. The information that holds the lowest uncertainty will be favored. Figure 4.1 presents the result of two fusions, each with a different Gaussian density being the most certain one.

4.2.1 Kalman Filter

The Bayesian recursive solution described by Proposition 4.2.1 enables the computation of the optimal estimation of state vector x_k. However, its implementation is impossible for practical applications, since it relies on mathematical integrations and as time evolves, the observation sequence grows indefinitely. The sequential algorithm proposed by (Kalman, 1960) solves that problem adding a restriction in the system
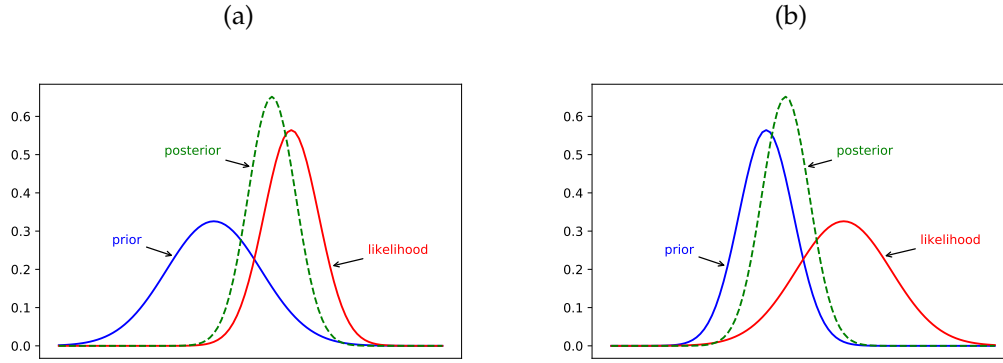
Figure 4.1: *Posterior* PDF obtained by the fusion of *prior* and *likelihood* densities. In (a) the variance of the *likelihood* is smaller than the variance of the *prior*, hence the *posterior* is closer to the *likelihood*. In (b) is the other way around.

assumptions, that is linearity and Gaussianity.

Consider the Gauss-Markov discrete-time linear system

$$x_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + G_{k-1}w_{k-1}, \tag{4.49}$$

$$y_k = C_k x_k + v_k, \tag{4.50}$$

where, $\forall k \geq 1$, time varying matrices $A_{k-1} \in \mathbb{R}^{n \times n}$, $B_{k-1} \in \mathbb{R}^{n \times p}$, $G_{k-1} \in \mathbb{R}^{n \times q}$ and $C_{k-1} \in \mathbb{R}^{m \times n}$ are known, as well as the input $u_{k-1} \in \mathbb{R}^{P}$ and output $y_k \in \mathbb{R}^m$ vectors. Process and observation noise vectors, $w_{k-1} \in \mathbb{R}^q$ and $v_{k-1} \in \mathbb{R}^m$ are white, zero-mean and mutually independent, apart from being Gaussian, with known covariance matrices Q_{k-1} and R_k, respectively.

Define $\mathcal{N}(x; \bar{x}, P^{xx})$ as a multivariate Gaussian PDF on x, with mean $\bar{x}$ and covariance P^{xx}, given by

$$\mathcal{N}(x; \mu, P^{xx}) \triangleq \frac{1}{(2\pi)^{n/2} |P^{xx}|^{1/2}} exp\left(-\frac{1}{2}(x - \mu)^T (P^{xx})^{-1}(x - \mu)\right) \tag{4.51}$$

Some identities of multivariate Gaussian probability densities are needed for the next steps and are described in Properties 4.2.1 and 4.2.2.

Property 4.2.1. If the random variables x and y are Gaussian RVs,

$$\rho(x,y) = \mathcal{N}\left(\begin{pmatrix} x \\ y \end{pmatrix}; \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}, \begin{pmatrix} P^{xx} & P^{xy} \\ (P^{xy})^T & P^{yy} \end{pmatrix}\right) \tag{4.52}$$

then, the marginal and conditional PDFs of x and y are given by

$$\rho(x) = \mathcal{N}(x; \bar{x}, P^{xx}) \tag{4.53}$$

$$\rho(y) = \mathcal{N}(y; \bar{y}, P^{xy}) \tag{4.54}$$

$$\rho(x|y) = \mathcal{N}(x; \bar{x} + P^{xy}(P^{yy})^{-1}(y - \bar{y}), P^{xx} - P^{xy}(P^{yy})^{-1}(P^{xy})^T) \tag{4.55}$$

$$\rho(y|x) = \mathcal{N}(y; \bar{y} + (P^{xy})^T(P^{xx})^{-1}(x - \bar{x}), P^{yy} - (P^{xy})^T(P^{xx})^{-1}P^{xy}) \tag{4.56}$$

$\square$

Property 4.2.2. If x and y are Gaussian RVs, with PDFs given by

$$\rho(x) = \mathcal{N}(x; \bar{x}, P^{xx}), \tag{4.57}$$

$$\rho(y|x) = \mathcal{N}(y; Ax, P^{yy}), \tag{4.58}$$

then, the joint and marginal PDFs are defined by

$$\rho(x,y) = \mathcal{N}\left(\begin{pmatrix} x \\ y \end{pmatrix}; \begin{pmatrix} \bar{x} \\ A\bar{x} \end{pmatrix}, \begin{pmatrix} P^{xx} & P^{xx}A^T \\ AP^{xx} & AP^{xx}A^T + P^{yy} \end{pmatrix}\right), \tag{4.59}$$

$$\rho(y) = \mathcal{N}(y; A\bar{x}, AP^{xx}A^T + P^{yy}) \tag{4.60}$$

$\square$

From the Gauss-Markov assumptions, we can rewrite (4.49) and (4.50) from a probabilistic perspective, as

$$\rho(x_k|x_{k-1}) = \mathcal{N}(x_k; A_{k-1}\bar{x}_{k-1} + B_{k-1}u_{k-1}, Q_k), \tag{4.61}$$

$$\rho(y_k|x_k) = \mathcal{N}(y_k; C_k\bar{x}_k, R_k), \tag{4.62}$$

where (4.61) is the *transition* density representing the system dynamics and (4.62) is the *likelihood* density, given by the observation model.

The Bayesian recursive solution to such system is defined by *forecast* and *data assimilation* steps according to

$$\rho(x_k|(y_1,\ldots,y_{k-1}) = \mathcal{N}(x_k; \hat{x}_{k|k-1}, P^{xx}_{k|k-1}), \tag{4.63}$$

$$\rho(x_k|(y_1,\ldots,y_k) = \mathcal{N}(x_k; \hat{x}_{k|k}, P^{xx}_{k|k}), \tag{4.64}$$

with the *posterior* density function from a previous step given by

$$\rho(x_{k-1}|(y_1,\ldots,y_{k-1}) = \mathcal{N}(x_{k-1}; \hat{x}_{k-1|k-1}, P^{xx}_{k-1|k-1}), \tag{4.65}$$

where $\hat{x}_{k|k-1}$ and $P^{xx}_{k|k-1}$ are the *forecast* state and covariance estimates, whereas $\hat{x}_{k|k}$ and $P^{xx}_{k|k}$ are the *data assimilation* state and covariance estimates.

Now we combine the *forecast steps* from (4.31) and from (4.63), using the process model from (4.61), the previous estimate from (4.65) and the identities from Property 4.2.2, yielding

$$\rho(x_k|(y_1,\ldots,y_{k-1})) = \int_{\mathbb{R}^n} \rho(x_k|x_{k-1})\rho(x_{k-1}|y_1,\ldots,y_{k-1})dx_{k-1}, \tag{4.66}$$

$$= \mathcal{N}(x_k; A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}, A_{k-1}P^{xx}_{k-1|k-1}Ak-1^T + G_{k-1}QG^T_{k-1}),$$

that is, the *forecast* state and covariance estimates are computed by

$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}, \tag{4.67}$$

$$P^{xx}_{k|k-1} = A_{k-1}P^{xx}_{k-1|k-1}A^T_{k-1} + G_{k-1}QG^T_{k-1}. \tag{4.68}$$

And for the *data assimilation step*, we find the joint density of y_k with the *forecast* estimate from (4.66), using Property 4.2.1

$$\rho(x_k,y_k|(y_1,\ldots,y_{k-1})) = \mathcal{N}\left(\begin{pmatrix} x_k \\ y_k \end{pmatrix}; \begin{pmatrix} \hat{x}_{k|k-1} \\ C_k\hat{x}_{k|k-1} \end{pmatrix}, \begin{pmatrix} P^{xx}_{k|k-1} & P^{xx}_{k|k-1}C^T_k \\ C_kP^{xx}_{k|k-1} & C_kP^{xx}_{k|k-1}C^T_k + R_k \end{pmatrix}\right), \tag{4.69}$$

and the marginal density for x_k is

$$\rho(x_k|(y_1,\ldots,y_k)) = \mathcal{N}(x_k;\ \hat{x}_{k|k-1} + K_k(y_k - C_k\hat{x}_{k|k-1}), P^{xx}_{k|k-1} - P^{xx}_{k|k-1}C_k^T K_k^{-1} P^{xx}_{k|k-1}C_k^T), \quad (4.70)$$

that is, the *data assimilation* state and covariance estimates are calculated by

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C_k\hat{x}_{k|k-1}), \quad (4.71)$$

$$P^{xx}_{k|k} = P^{xx}_{k|k-1} - P^{xx}_{k|k-1}C_k^T K_k^{-1} P^{xx}_{k|k-1}C_k^T. \quad (4.72)$$

where $K_k \in \mathbb{R}^{n \times m}$ is defined as the Kalman gain and is given by

$$K_k = P^{xx}_{k|k-1}C_k^T (C_k P^{xx}_{k|k-1} C_k^T + R_k)^{-1}. \quad (4.73)$$

If we define a *forecast* output estimate $\hat{y}_{k|k-1}$, an innovation covariance matrix $P^{yy}_{k|k-1}$ and a cross-covariance matrix $P^{xy}_{k|k-1}$ as

$$\hat{y}_{k|k-1} \triangleq C_k\hat{x}_{k|k-1}, \quad (4.74)$$

$$P^{yy}_{k|k-1} \triangleq E\left[(y_k - \hat{y}_{k|k-1})(y_k - \hat{y}_{k|k-1})^T\right] = C_k P^{xx}_{k|k-1} C_k^T + R_k, \quad (4.75)$$

$$P^{xy}_{k|k-1} \triangleq E\left[(x_k - \hat{x}_{k|k-1})(y_k - \hat{y}_{k|k-1})^T\right] = P^{xx}_{k|k-1} C_k^T, \quad (4.76)$$

$$(4.77)$$

we can simplify *data assimilation* step given by (4.71), (4.72) and (4.73) as

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}), \quad (4.78)$$

$$P^{xx}_{k|k} = P^{xx}_{k|k-1} - P^{yy}_{k|k-1} K_k^{-1} (P^{yy}_{k|k-1})^T, \quad (4.79)$$

$$K_k = P^{xy}_{k|k-1} (P^{yy}_{k|k-1})^{-1}. \quad (4.80)$$

For more details on this derivation, refer to (Särkkä, 2013). We summarize the Kalman filter algorithm as below

Algorithm 4.2.1. Kalman filter (KF) algorithm.

Forecast step: Using the linear model, calculate the estimated state vector $\hat{x}_{k|k-1}$ and the estimated state covariance $P^{xx}_{k|k-1}$, using (4.67) and (4.68). Calculate Kalman gain K_k by (4.73).

Data assimilation step: Using the measurement vector y_k and Kalman gain, update estimations from previous step with (4.71) and (4.72), obtaining estimates $\hat{x}_{k|k}$ and $P^{xx}_{k|k}$.

$\square$

The linear KF is the optimal estimator under both MAP and MMSE criteria, it is unbiased and its cross-covariance will asymptotically achieve the lower bound of the Cramér-Rao inequality (Teixeira, 2008).

4.2.2   Unscented Kalman Filter

For nonlinear systems, the Gaussianity requirement does not hold, even if the uncertainty and initial conditions are Gaussian. Therefore, we can not characterize the posterior PDF only by its first two moments, mean and covariance, and the solution given by proposition 4.2.1 are not suited for the estimation problem. In fact, optimal solutions are generally not possible to be obtained by a recursive algorithm.

Adaptations to the Kalman filter have been proposed in the literature. The extended Kalman filter (EKF), for example, linearizes the system around the current state estimates, to achieve approximations using Algorithm 4.2.1. A different approach is to approximate the statistics of the posterior PDF, instead of the model. The unscented Kalman filter (UKF) (Julier and Uhlmann, 2004) performs statistical approximations via the unscented transform (UT), based on the Monte Carlo approach.

First, we define a set of samples $\chi \triangleq [\chi_1, \ldots, \chi_{2n}] \in \mathbb{R}^{n \times 2n}$, also referred to as sigma points, to estimate the mean $\bar{x}$ and covariance $\bar{P}^{xx}$ of the random variable being approximated, such that

$$\sum_{i=0}^{2n} \gamma_i \chi_i = \bar{x}, \tag{4.81}$$

$$\sum_{i=0}^{2n} \gamma_i \left[(\chi_i - \bar{x})(\chi_i - \bar{x})^T \right] = \bar{P}^{xx}, \tag{4.82}$$

$$\sum_{i=0}^{2n} \gamma_i = 1. \tag{4.83}$$

where γ_i, $\forall i > 1$ are called weights. There are various ways of determining the sigma points, but we choose a simplified version of the original UT proposed by (Julier and Uhlmann, 1997), which each χ_i is given by a column of the matrix $\chi = [\chi_1\ \chi_2\ ...\ \chi_{2n}]$ given by

$$\chi \triangleq \left[\bar{x} 1_{1\times 2n} + \sqrt{n}(\bar{P}^{xx})^{1/2} \quad \bar{x} 1_{1\times 2n} - \sqrt{n}(\bar{P}^{xx})^{1/2} \right], \tag{4.84}$$

with weights

$$\gamma = \frac{1}{2n}. \tag{4.85}$$

Such simplification has one less tunning degree of freedom to approximate higher order moments of the RV, called scaling factor. We now define the UT algorithm as follows

Algorithm 4.2.2. Unscented Transform (UT) algorithm
Consider a nonlinear transformation of two random variables y and x

$$y = f(x,d), \tag{4.86}$$

where x has mean $\bar{x} \in \mathbb{R}^n$ and covariance matrix $P^{xx} \in R^{n\times n}$ and d is a deterministic vector. First, compute sigma points matrix χ and its correspondent weights γ according to (4.84) and (4.85) respectively. Propagate each sigma point χ_i through the nonlinear transformation given by (4.86) to find estimated mean $\hat{y}$ and covariance matrix P^{yy} of the RV y and the cross-covariance matrix P^{xy}, by

$$\Upsilon_i = f(\chi_i, d), \quad \forall i = 1, \dots, 2n \tag{4.87}$$

$$\hat{y} = \sum_{i=0}^{2n} \gamma_i \Upsilon_i, \tag{4.88}$$

$$P^{yy} = \sum_{i=0}^{2n} \gamma_i \left[(\Upsilon_i - \hat{y})(\Upsilon_i - \hat{y})^T \right], \tag{4.89}$$

$$P^{xy} = \sum_{i=0}^{2n} \gamma_i \left[(\Upsilon_i - \hat{y})(\chi_i - \bar{x})^T \right]. \tag{4.90}$$

For simplicity, we will refer to (4.87-4.90) as the function ψ_{UT}, given by

$$\left[\hat{y} \ \ P^{yy} \ \ P^{xy} \right] = \psi_{UT} \left(\bar{x}, P^{xx}, n, d, f \right) \tag{4.91}$$

$\square$

Based on Algorithm 4.2.2, and considering the process and observation models given by (1.1) and (1.2), the UKF algorithm is as follows

Algorithm 4.2.3. Unscented Kalman filter (UKF) algorithm.
Forecast step: Use the UT to calculate the forecast state estimate $\hat{x}_{k|k-1}$ and the respective covariance $P^{xx}_{k|k-1}$ based on last estimation

$$\left[\hat{x}_{k|k-1} \ \ P^{xx}_{k|k-1} \right] = \psi_{UT} \left(\hat{x}_{k-1|k-1}, P^{xx}_{k-1|k-1}, n, d, f \right), \tag{4.92}$$

where f is given by (1.1) and d by current input data and time instant.
Based on forecast state estimate and covariance found in (4.92), compute the forecast observation estimate $\hat{y}_{k|k-1}$, observation vector covariance $P^{yy}_{k|k-1}$ and the cross-covariance between observation and state $P^{xy}_{k|k-1}$, via the UT

$$\left[\hat{y}_{k|k-1} \ \ P^{yy}_{k|k-1} \ \ P^{xy}_{k|k-1} \right] = \psi_{UT} \left(\hat{x}_{k|k-1}, P^{xx}_{k|k-1}, n, d, g \right), \tag{4.93}$$

where g is given by (1.2) and d by current time instant.
Data assimilation step: Calculate Kalman gain and current state estimate and covariance

at time k, same way as in (4.78-4.80), reproduced for convenience

$$K_k = P^{xy}_{k|k-1}(P^{yy}_{k|k-1})^{-1}, \tag{4.94}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}), \tag{4.95}$$

$$P^{xx}_{k|k} = P^{xx}_{k|k-1} - P^{yy}_{k|k-1}K_k^{-1}(P^{yy}_{k|k-1})^T. \tag{4.96}$$

$\square$

UKF algorithm addresses many of the drawbacks that appear in the EKF implementation, such as the necessity of differentiability, the performance loss due to systems that are poorly approximated by linearization and computational efficiency. Therefore the UKF is the chosen method for the nonlinear system simulated in Chapter 5 and will be further discussed in this section.

4.3 State Estimation with Aperiodic Sampling

The exact solution to the linear discretization problem, defined by (4.11)-(4.16) considered digital inputs sampled at the same rate as the observations are taken, producing a sequence of data at time instants t_k, $\forall k \in \mathbb{N}$. The nonlinear solution is agnostic to this assumption, since it relies on numerical approximations. Nevertheless, the representation given by (4.9) and (4.10) is constructed using a common time sequence t_k, $\forall k \in \mathbb{N}$ for both process and observation models.

However, we are interested in the effects of aperiodic sampling of the observations with data assimilation executed at incorrect time instants, as formulated in Section 1.2. For that, we consider that input signal $u(t)$ is generated by a digital device using an *ideal sampler*, with a regular time interval T, that is $u(t) = u(iT)$, for $iT \leq t < (i+1)T$, $\forall i \in \mathbb{N}$. Estimation time instants need to coincide with the regularly spaced time intervals T. Measurement instants t_k, $\forall k \in \mathbb{N}$ do not match the estimation instants iT, $\forall i \in \mathbb{N}$. Additionally, input data $u(iT)$ is available a rate $1/T$ that is $\alpha > 1$ times faster than the expected sampling rate λ of the observations. An illustrative example is presented in Figure 4.2, where $\alpha = 5$, that is the expected time interval for observations $1/\lambda$ is five times higher than regular sampling time interval T.

Between two consecutive estimation time instants, there can be no observation data, illustrated by the red letter A, there can be one measurement, red letter B, or
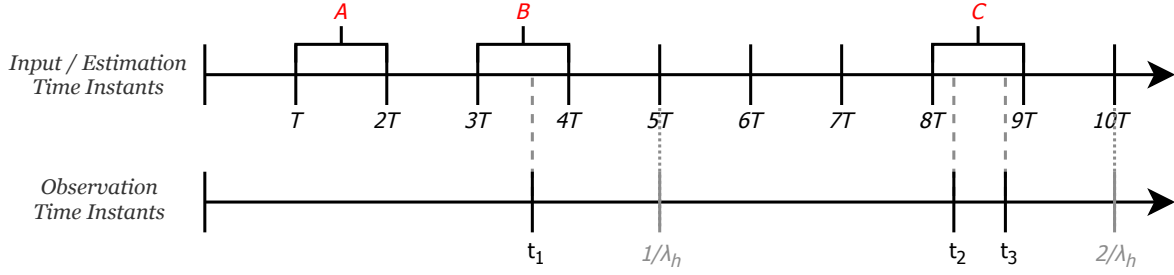
Figure 4.2: Input and estimation time instants labeled as iT and a realization of observation time instants, labeled as t_k. Expected time instants for observation is presented in gray, considering that $E[h_k] = \frac{1}{\lambda}$, and $\frac{1}{\lambda} = 5T$, that is $\alpha = 5$.

multiple measurements, shown in the red letter C. One example of such application is in target tracking, where inertial sensors that measure process input data such as linear acceleration and angular velocity operate at higher and regular frequencies compared to a network of GPS sensors, that are responsible for the observation model data.

On the next subsections, we present the motivation behind the aperiodic sampling model and discuss how the algorithm is executed for the scenarios when reliable time-stamp information is available, and when it is not part of the data package.

4.3.1   Aperiodic Sampling as a Poisson Process

In Section 1.2 we assume the observation time instants t_k occur randomly in time according to a stationary Poisson process. That is, the probability of observing $N(t) = n$ measurements up to and including time t is given by (Papoulis, 1984)

$$P\left(N(t) = n\right) = e^{-\lambda t}\frac{(\lambda t)^n}{n!}, \tag{4.97}$$

where λ is called rate or intensity parameter, $t = t_1 - t_0$ is the time difference between two points and $N(t)$ is a RV representing the amount of measurements that arrived in the interval given by t, as shown in Figure 4.3. The interarrival time h_k, given by the distance between two consecutive arrivals, for the Poisson process has an exponential distribution, with a PDF given by

$$\rho_{h_k}(t) = \lambda e^{-\lambda t} \tag{4.98}$$

This assumption is proposed by (Micheli and Jordan, 2002), motivated by sensor networks applications. Consider the LTI system described by (4.11) and (4.12). Now
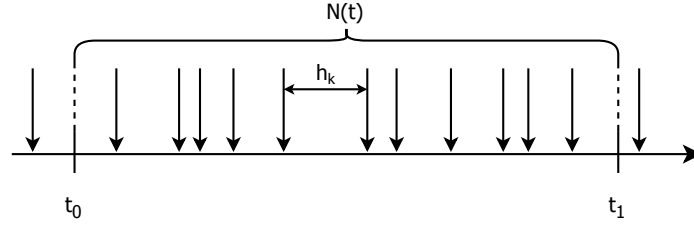
Figure 4.3: Arrivals of a Poisson process between a time interval given by $t = t_1 - t_0$. $N(t)$ is the RV representing the discrete amount of arrivals in the time interval and h_k is the distance between two consecutive arrivals.

assume N sensors measure such system periodically, every T seconds, according to

$$y_i(nT + \xi_i) = Cx(nT + \xi_i) + v(nT + \xi_i), \quad n \in \mathbb{N}, i \in \{1,2,\ldots,N\}, \tag{4.99}$$

where matrix C and measurement noise are already defined in (4.12) and $\xi_i \in [0,T)$ is referred to as the phase of each sensor, that is the waiting time for the i-th sensor to yield a noisy measurement, after time $t = 0$. If all phases are different, that is the sensors are not synchronized, and the number of sensors N is high enough, then the amount of measurements arriving in a given interval can be approximated by a Poisson process with rate parameter $\lambda = N/T$. Thus, the random interarrival times or the waiting times between two consecutive measurements can be approximated by an exponential random variable $h_k \sim \mathcal{E}(\lambda)$, according to Proposition 4.3.1.

Property 4.3.1. The cumulative distribution function (CDF) of a random variable X is given by (Papoulis, 1984)

$$F_X(x) = \rho(X \le x), \tag{4.100}$$

where $\rho(X \le x)$ denotes the probability of that RV X takes on values smaller or equal to x. The complementary CDF is defined by

$$\bar{F}_X(x) = 1 - \rho(X > x). \tag{4.101}$$

$\square$

Property 4.3.2. The cumulative distribution function (CDF) of a uniform random vari-

able X in the interval $[a,b]$ is given by (Papoulis, 1984)

$$F_X(x) = \begin{cases} 0, & \text{if } x < a, \\ \frac{x-a}{b-a}, & \text{if } a \leq x \leq b, \\ 1, & \text{if } x > b. \end{cases} \tag{4.102}$$

□

Property 4.3.3. The cumulative distribution function (CDF) of an exponential random variable X with rate parameter λ is given by (Papoulis, 1984)

$$F_X(x; \lambda) = \begin{cases} 0, & \text{if } x < 0, \\ 1 - e^{-\lambda x}, & \text{if } x \geq 0, \end{cases} \tag{4.103}$$

□

Proposition 4.3.1. Let the phases of all sensors ξ_i, for $i = 1,2,\ldots N$ be i.i.d uniform random variables in the interval $[0,N/\lambda]$, with $\lambda \in \mathbb{R}$ and $N \in \mathbb{N}$. Now define $h_N = \min(\xi_1,\xi_2,\ldots \xi_N)$ as the first interarrival time, occurring in $[0,N/\lambda]$. Then, the RV h_N converges in distribution to an exponential RV, that is

$$h_N \xrightarrow{D} \mathcal{E}(\lambda), \tag{4.104}$$

where $\xrightarrow{D}$ means *converges in distribution*, as $N \to \infty$.

□

Proof. The distribution of h_N is studied by the so called extreme value theory (Leadbetter et al., 1983). To compute it, we consider its cumulative distribution function (CDF), the independence condition and Property 4.3.1, yielding

$$\rho_{h_N}\left(\min_i(\xi_i) \le t\right) = 1 - \rho\left(\min_i(\xi_i) > t\right),$$

$$= 1 - \rho\left(\xi_1 > t, \xi_2 > t, \ldots, \xi_N > t\right),$$

$$= 1 - \prod_{i=1}^{N} \rho(\xi_i > t),$$

$$= 1 - \prod_{i=1}^{N} \left[1 - \rho(\xi_i \le t)\right],$$

$$= 1 - \left[1 - \rho(\xi_i \le t)\right]^N. \tag{4.105}$$

But since ξ_i for $i = 1,2,\ldots N$ are i.i.d. uniform random variables in the interval $[0,N/\lambda]$, we can rewrite 4.105 using Property 4.3.2, as

$$\rho_{h_N}\left(\min_i(\xi_i) \le t\right) = 1 - \left(1 - \frac{\lambda t}{N}\right)^N,$$

$$\to 1 - e^{-\lambda t}, \quad \text{as } N \to \infty. \tag{4.106}$$

Thus, from Property 4.3.3, we conclude that h_N tends, in distribution, to an exponential random variable with rate parameter λ, which proves Proposition 4.3.1.

$$\square$$

If we define $\lambda = N/T$, with a known time interval T and a sufficiently big N amount of sensors, then we can approximate all interarrival times as an exponential RV of rate parameter λ, since the exponential distribution has the memoryless property. Hence the arrival sequence can be approximated by a Poisson process.

4.3.2 State Estimation With Timestamp

If the estimator knows the exact time t_k that measurements were taken in a global timescale, *data assimilation* steps can be performed at the correct time instants. For that, the process model discretization given by (4.9) is performed considering variable time intervals δt_j^*, yielding

$$x(t_j^*) = f_d(x(t_{j-1}^*), u(t_{j-1}^*), w(t_{j-1}^*), t_{j-1}^*), \tag{4.107}$$

where t_j^*, $\forall j \in \mathbb{N}$ represents the estimation time sequence and is given by $t_j^* = t_{j-1}^* + \delta t_j^*$, with $t_0^* = 0$. The variable time interval δt_j^* is calculated according the schematic presented by Figure 4.4.
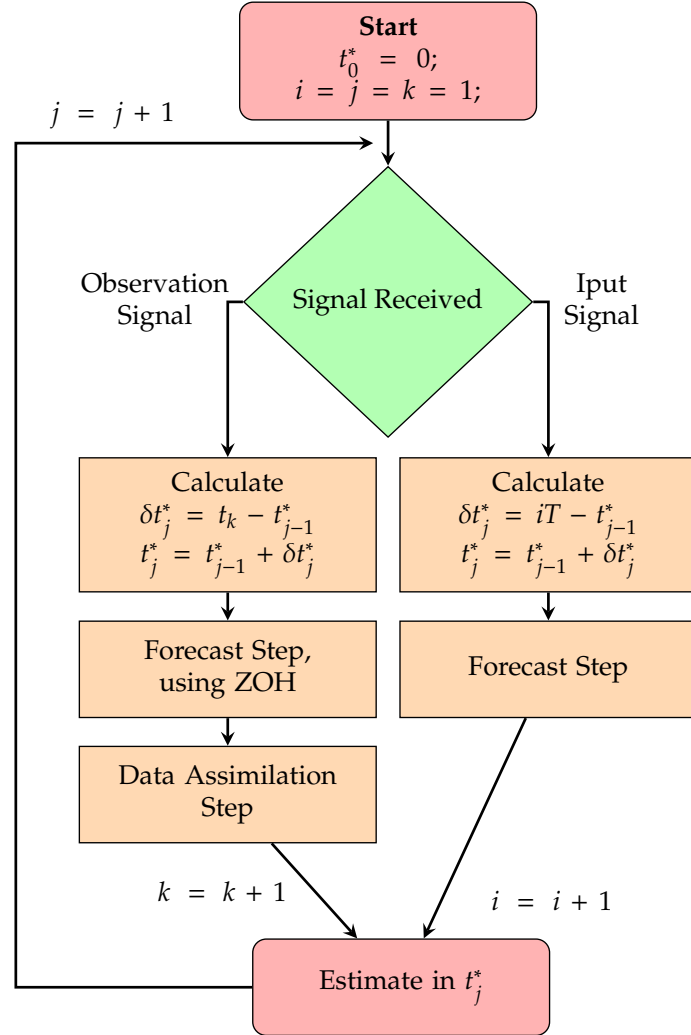
To analyze

Figure 4.4: Illustrative schematic of the *online* estimator, with time-stamp. indices i, j e k represent the input, estimation and observation signal counters, respectively.

Each value δt_j^* corresponds to the time interval between the last instant t_{j-1}^* in which a signal was received, whether it transmitted input or observation data, and the next time interval t_j^* in which a new signal arrives. A zero order holder (ZOH) is used for input signals, considering last available information.

For the simulation carried out in Section 5, time intervals δt_j^* are calculated by uniting all arrival times for input and observation in a single vector, in an orderly fashion. The

subtraction of consecutive time instants yields the time intervals sequence δt_j^*, $\forall j \geq 1$.

Since there are two signal types, input and observation, there are four possible cases for the estimator: input followed by another input; input followed by an observation; observation followed by an input; and observation followed by another observation. In Figure 4.2, all of them are represented. During the interval A, there are only input signals, so time interval is T and only *forecast* is performed. In the interval represented by B, we have to first execute complete *forecast* and *data assimilation* steps between input and observation, from $3T$ until t_1, using $\delta t_4^* = t_1 - 3T$. Next, between an observation and an input signal, a ZOH is used for a *forecast* step between t_1 and $4T$. In other words, it is considered that the input remained constant between $3T$ and $4T$. When more than one observation arrive between two input signals, as in C, full *forecast* and *data assimilation* are performed as many times needed before one last *forecast* between the last observation and the next input signal.

For the *online* estimator, the differential equations are numerically integrated as input or measurement signals arrive. In these instants, the corresponding time interval δt_j^* is calculated. If the current signal transmits input data, only the *forecast* step is executed. If it is observation data being received, both *forecast* and *data assimilation* steps are performed, considering a ZOH for the input data.

4.3.3 State Estimation Without Timestamp

For some applications, discretization at variable time intervals may be difficult to accomplish. Or measurements may be transmitted in a centralized manner at regular time intervals, even though they were taken aperiodically. In such cases it is impossible to consider the time-stamp information in the estimation process. Thus process model (1.1) is discretized according to

$$x_n = f_d^*(x_{n-1}, u_{n-1}, w_{n-1}, n), \tag{4.108}$$

where $f_d^*(\cdot)$ represents the time-invariant discrete-tmie mode and $n = nT$, $\forall n \in \mathbb{N}$.

Since the estimator is not aware of the measurement time instant, *data assimilation* is always performed as the next input signal arrives, in a time instant multiple of T. In other words, the observation vector is approximated by $\tilde{y}_i \approx y(t_k)$, where y_i is the measurement taken at the instant t_k, assimilated at time instant iT, for $(i-1)T < t_k \leq iT$ and t_k being the instant of the most recent measurement in the interval $[(i-1)T, iT]$.

Thus there are only two possible estimation scenarios. One, in which there are no information between two consecutive input signal arrivals, represented by the letter A in Figure 4.2, when only *forecast* step is performed. If there is at least one observation, a complete estimation with *forecast* and *data assimilation* steps is performed, considering the time interval T. In case illustrated by the letter B, measurement taken at time t_1 will be assimilated in time instant $4T$. When there are multiple measurements between two input signals, the oldest ones are discarded, as in letter C, for which the measurement taken at t_2 is discarded and the one taken at t_3 is assimilated at the instant $9T$.

Example 4.3.1. Consider the LTI system, represented by the following transfer function

$$G(s) = \frac{100}{s^2 + 2s + 100},\tag{4.109}$$

whose continuous-time state space model can be represented by

$$\dot{x}(t) = \begin{bmatrix} -2 & -12.5 \\ 8 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 4 \\ 0 \end{bmatrix} u(t),\tag{4.110}$$

$$y(t) = \begin{bmatrix} 0 & 3.125 \end{bmatrix} x(t).\tag{4.111}$$

A pseudo-random binary sequence (PRBS) with 200 samples is generated and oversampled by a factor of 100, resulting in 2×10^4 samples over 2 seconds, and the simulations results are shown in Figure 4.5.

Now we simulate the aperiodic sampling of the output $y(t)$ as a Poisson process with rate $\lambda = 100$ Hz and compare it to the regular sampling with frequency $f = 100$ Hz. Figure 4.6 presents the comparison.

For such an example, considering an input signal sampled at a frequency of $f = 100$ Hz, the random measurement time instants of the algorithm without timestamp are approximated to the next instant multiple of $T = 0.001$ s. An illustration of such process is presented in Figure 4.7.

$\square$

This approximation process leads to two sources of degradation. First, some information will be lost, if there are more than one observations in the interval $[iT,(i+1)T]$ $\forall i \in \mathbb{N}$. The second one is related to the measurement error of shifting time instants. A more detailed analysis of the latter effect is discussed in the Section 5.1.
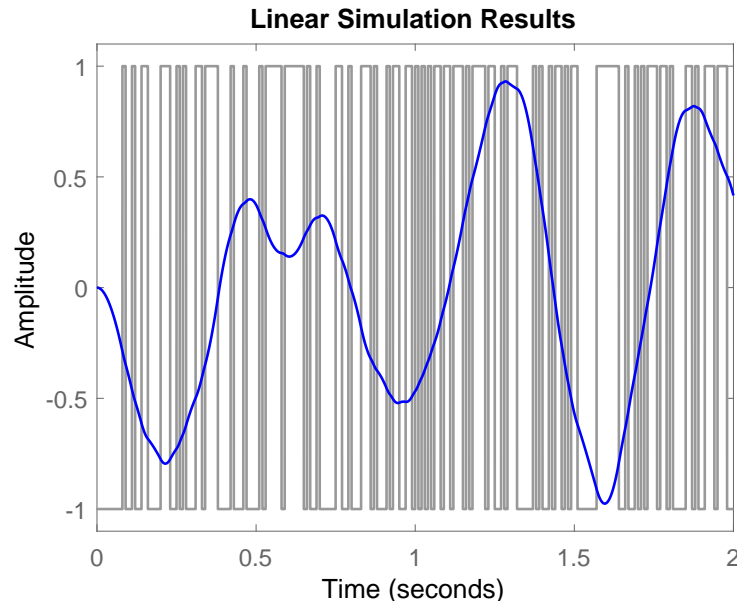
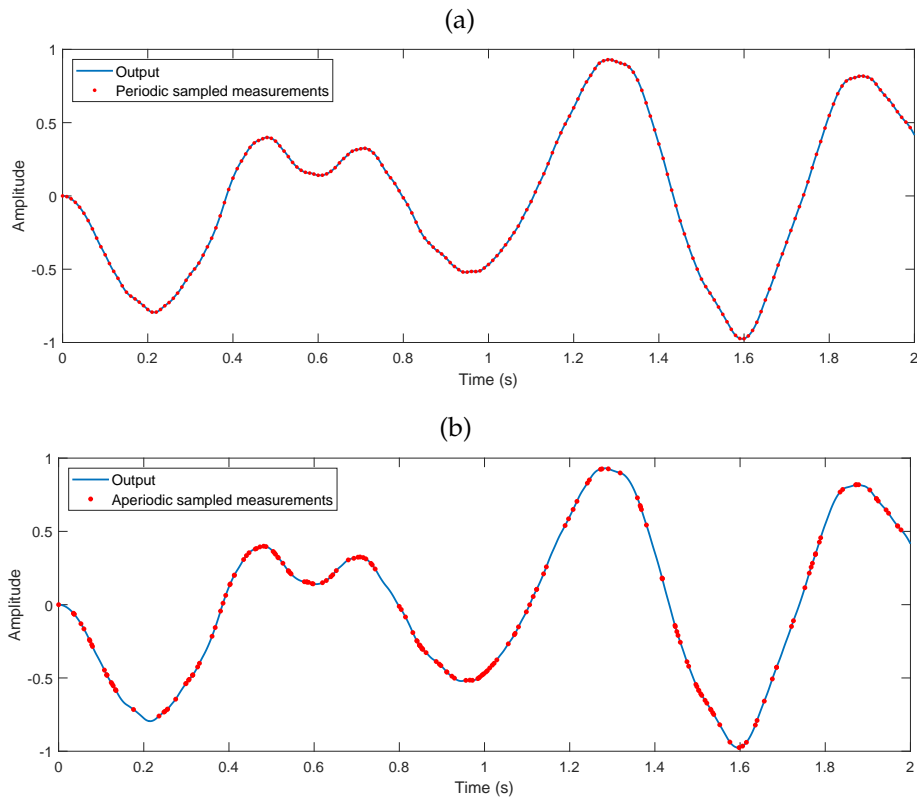Figure 4.5: Linear simulation results for G(s) in blue, considering the PRBS input in grey.

Figure 4.6: Periodic (a) and aperiodic (b) sampling of the output variable of G(s). Aperiodic sampling instants sequence generated by a Poisson process with intensity $\lambda = 100$ Hz.
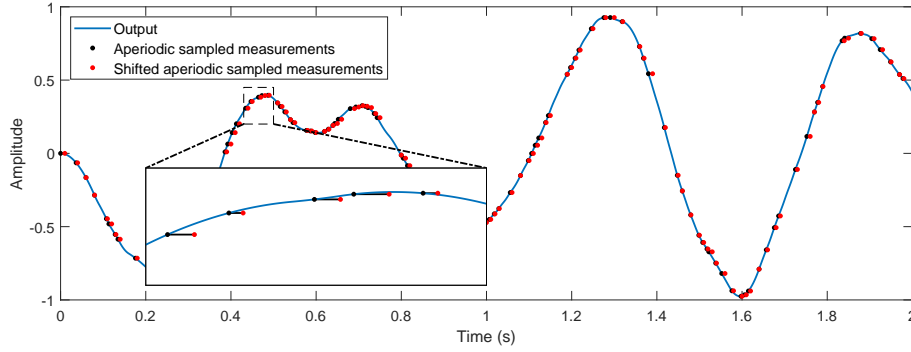
Figure 4.7: Effect of approximating the random time instants of the aperiodic sampling. Black dots are the true measurements and the red dots are the measurements shifted to the next multiple of $T = 0.001$.

4.4 Performance Metrics

In order to assess the performance degradation introduced by assimilating data at incorrect time instants, we need to define certain performance metrics for comparison. The algorithms described in Sections 4.2.2 and 4.2.1 estimate both the current state, $\hat{x}_{k|k}$ and its covariance matrix $\hat{P}^{xx}_{k|k}$. For the linear case, the *posterior* conditional PDF $\rho(x_k|(y_1,\ldots,y_k)$ is Gaussian, according to (4.64), so it is fully characterized by its first two moments. Thus, if the filter is consistent, the following conditions shall be met (Bar-Shalom et al., 2001)

$$E\left[x_k - \hat{x}_{k|k}\right] \triangleq E[\tilde{x}_{k|k}] = 0, \tag{4.112}$$

$$E\left[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T\right] \triangleq E[\tilde{x}_{k|k}\tilde{x}^t_{k|k}] = P^{xx}_{k|k}, \tag{4.113}$$

where $\tilde{x}_{k|k}$ is the estimation error at time instant k. Condition (4.112) is called *unbiasedness* requirement, whereas (4.112) refers to *covariance matching*. For the nonlinear case, these conditions cannot be fully met, since the *posterior* PDF is an approximation of a Gaussian density. Thus, the closer they are met, the more consistent are the filter estimates. In this study we will use metrics that measure both consistency conditions. According to Bar-Shalom, in order to test them, the consistency criteria metrics for state estimation must certify: that state estimate errors are zero mean and compatible with the estimated state covariance; that innovations are also zero mean and compatible with their respective covariances; and that innovations are white. We will adopt two tests proposed by him

that attests all conditions simultaneously, that is the normalized (state) estimation error squared (NEES) and normalized innovation squared (NIS) tests. We first defined NEES and NIS as

$$NEES_k \triangleq \tilde{x}_{k|k}^T (P_{k|k}^{xx})^{-1} \tilde{x}_{k|k}, \tag{4.114}$$

$$NIS_k \triangleq \eta_{k|k-1}^T (P_{k|k-1}^{yy})^{-1} \eta_{k|k-1}. \tag{4.115}$$

Under the linear and Gaussian assumption, we formulate a hypothesis test, under which the null hypothesis H_0, that the filter is consistent, requires that both NEES and NIS follow chi-squared distributions, with n_x and n_y degrees of freedom, respectively, and n_x is the dimension of the state vector, whereas n_y is the dimension of the observation vector. The expected value of a RV that is chi-squared distributed is equal to its degrees of freedom quantity, that is $E[NEES_k] = n_x$, $\forall k > 1$ and $E[NIS_k] = n_y$, $\forall k > 1$.

We adopt single-run tests, thus for every estimation, we test the acceptance of H_0, that is if both NEES and NIS at time instants k are within a certain interval, considering the accepted region as

$$P\{NEES_k \in [r_1, r_2] | H_0\} = 1 - \alpha, \tag{4.116}$$

$$P\{NIS_k \in [r_1, r_2] | H_0\} = 1 - \alpha, \tag{4.117}$$

where α is the significance level and interval $[r_1, r_2]$ is given by the chi-square distribution degrees of freedom and α. For instance, considering a system with a state vector of size 4 and observation vector of size 2, and a significance level of $\alpha = 5\%$, the acceptance intervals for NEES and NIS are respectively given by

$$\left[\chi_2^2(0.025), \chi_2^2(0.975) \right] = [0.051, 7.38], \tag{4.118}$$

$$\left[\chi_4^2(0.025), \chi_4^2(0.975) \right] = [0.484, 11.1], \tag{4.119}$$

which means that, if the filter is consistent, in 95% of the estimates, NEES and NIS value should fall under their correspondent intervals. Figure 4.8 shows both χ_2^2 and χ_4^2 PDFs, with the 95% acceptance region.
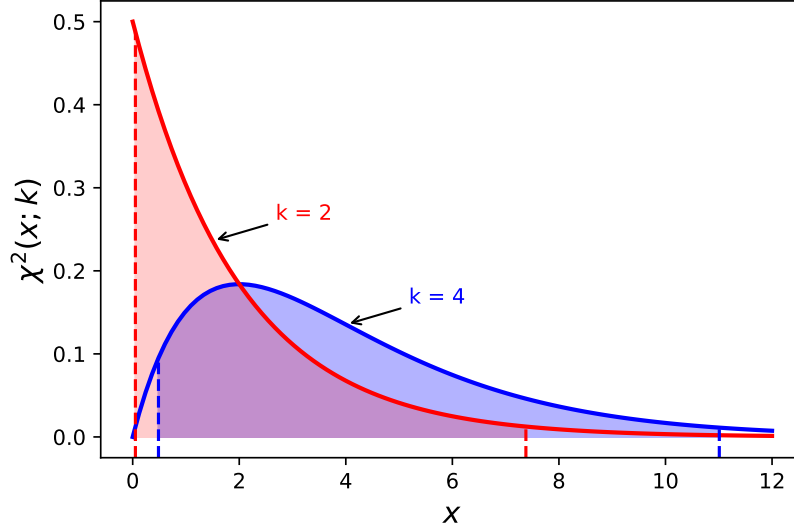
Figure 4.8: χ^2 PDFs for 2 (red) and 4 (blue) degrees of freedom. The vertical dashed lines represent the acceptance interval limits, for $\alpha = 5\%$, and the shaded area is the corresponding acceptance regions.

Additionally, since we are using simulated systems, the root mean square error (RMSE) of the system states will also be calculated as an accuracy performance index, given by

$$RMSE = \frac{\sum_{i=1}^{N} \sqrt{(\hat{x}_{k|k} - x_k)^2}}{N} \tag{4.120}$$

where x_k is the true state vector and N is the amount of estimates.

Numerical Results

In this chapter, we present simulation results of state estimation for two different sampled-data systems: an arbitrary linear system with two underdamped modes and a nonholomonic unicycle position system. Both systems are described in state-space representations. We also present three simulation scenarios designed to enable the evaluation of the effects of aperiodic sampling in state estimation when we neglect information about the time-stamps.

Since simulations are carried out in digital computers, it is impossible to simulate the continuous-time variables of the sampled-data systems. Therefore, we try to reproduce the actual values of the states in a continuous manner, by choosing very small time steps δt_{sim} to simulate the nominal system model. The continuous random time intervals h_k, $k \in \mathbb{N}$ are generated by the exponential probability distribution function from Matlab and resulting time instants t_k are then approximated to the nearest discrete time instant, among all available samples.

5.1 Error Effects of Shifting Measurements Time Instants

There are interesting aspects of the error that is added by neglecting timestamp information and approximating the measurement time instant to the next multiple of the regular time interval T. In Figure 4.6 we observe how the error is always as the inverse of the derivative of the output evaluated at the measurement is taken. In Figure 5.1 we present the time series of such error overlapping the output. Additionally, the magnitude of the error is proportional to the magnitude of the derivative.

We simulate new data, considering a much faster rate parameter of the Poisson process that generates the aperiodic sampling time instants, $\lambda = 2$ kHz. The derivative of the signal is calculated, by multiplying $G(s)$ (4.109) by s and simulating the output
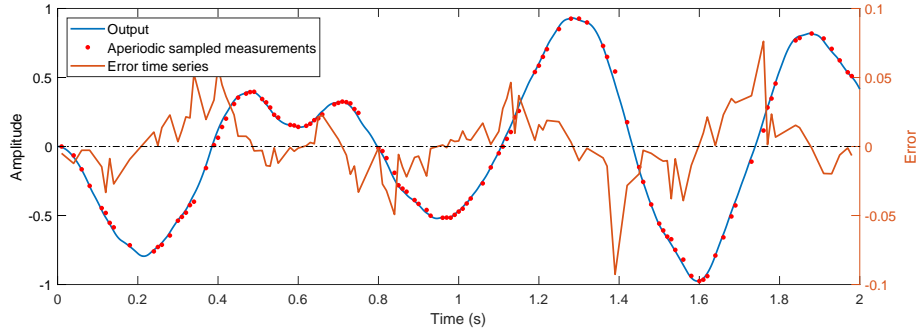
Figure 5.1: Added error time series (orange) as an effect of shifting time instants of the measurements (red) and output (blue). A gray dotted horizontal line is placed at the origin, for reference.

of the resulting system for the same input sequence. The original output, its derivative and the error time series are shown in Figure 5.2. We can see that the error values are directly proportional to the signal derivative. Therefore, one conclusion we can draw is that the error cannot be modeled as a white Gaussian noise, since it has strong correlation with consecutive points. In fact, from the quantile-quantile plot shown in Figure 5.3 we observe shows a strong deviation from the quantiles of the error data in comparison to those from the theoretical normal distribution.

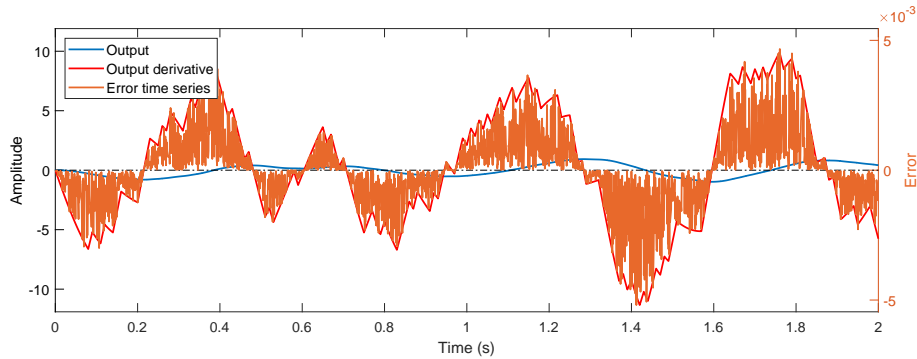Figure 5.2: Added error time series (orange) as an effect of shifting time instants of the measurements (red) and output (blue). A gray dotted horizontal line is placed at the origin, for reference.

5.2 Design of Simulation Setup

To evaluate performance and consistency effects of the aperiodic sampling in state estimation, we study three scenarios: variation of noise level in signals, or SNR; varia-
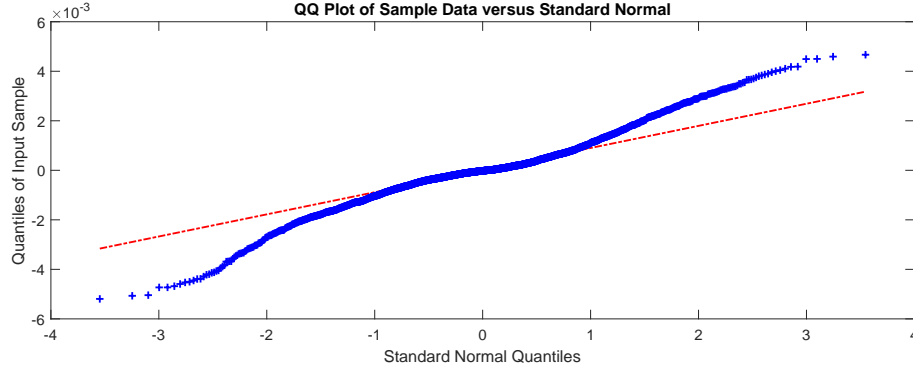
Figure 5.3: QQ-plot of the measurement error. Red line represents the expected quantiles of the standard normal distribution and the blue crosses, the error data quantiles.

tion of average measurement sampling rate λ; and variation of the relation between λ and the regular estimation sampling rate $1/T$, that is α.

In the first scenario we intend to study how performance is affected by the presence of different noise levels in the signals. It is expected that the higher the noise, the closer will be performance of both algorithms. That is because the additional noise tuned by not considering time-stamp information will be small in comparison to larger data noise level. The second scenario assesses how the average sampling rate of measurements is related to degradation in performance when time-stamp is not considered in state estimation. Therefore, the speed of system dynamics in comparison to the sampling rate is investigated. In the last simulation scenario we study the behavior of the effects when the regular estimation time interval distances itself from the average time interval of the observations. Additional errors due to measurement time instants approximation are expected to influence performance results.

Since estimation is performed in the presence of random quantities, then several runs of simulations need to be carried out, so that confidence intervals can be constructed towards more consistent conclusions. We calculate average values from all runs with the corresponding 95% confidence intervals for accuracy index *RMSE*, given by (4.120), and consistency tests *NEES*, given by (4.114) and (4.116), and *NIS*, given by (4.115) and (4.117). In Table 5.1 we review some of the concepts and symbols that are extensively used in this chapter

Table 5.1: Review of concepts and symbols used in simulation

Concept / Symbol	Definition	Description
λ (Section 1.2)	$h_k \sim \mathcal{E}(\lambda)$	Average sampling rate of observations, in Hertz, determining random time intervals h_k
T (Section 1.2)	$t_i \triangleq iT$	Regular estimation time interval, given in seconds
α (Section 1.2)	$\frac{1}{\lambda} \triangleq \alpha T$	Relation between regular estimation time interval T and average irregular time interval of observations $1/\lambda$
SNR (4.25)	$SNR_{\mathrm{dB}} \triangleq 10 \log_{10} \frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}$	Ratio between signal power P_{signal} and noise power P_{noise} in decibel, related to the level of data uncertainty
NEES (4.114)	$NEES \triangleq \tilde{x}^T (P^{xx})^{-1} \tilde{x}$	Value tested to be chi-squared distributed, in case of estimate error consistency
NIS (4.115)	$NIS \triangleq \eta^T (P^{yy})^{-1} \eta$	Value tested to be chi-squared distributed, in case of innovation consistency

5.3 Linear System

Our first system of choice is a fourth-order LTI system, since its results are more controllable and difference equations calculations have an exact solution. Thus results comparison and assessment are expected to be more clear and comprehensive. Additionally, we choose the combination of two independent modes, each of second order and with distinct dynamics, so the effects can be evaluated separately.

5.3.1 System Description

The linear system chosen for simulation is the serial combination of two second order underdamped modes, with different band pass behaviors. Figure 5.4 shows the bode diagrams of both systems separately. One of the modes, henceforth termed as low-pass (lp) system, has a time constant $\tau_{lp} = 1$ s, a natural frequency $\omega_{n,lp} = 10$ Hz and a damping constant of $\zeta_{lp} = 0.1$. The frequency response is then given by

$$G_{lp}(s) = \frac{100}{s^2 + 2s + 100}. \tag{5.1}$$

The second mode is a high-pass (hp) system, with a lower time constant $\tau_{hp} = 0.01$ s, a higher natural frequency $\omega_{n,hp} = 1000$ Hz, and same damping constant of $\zeta_{hp} = 0.1$, with two zeros, one at the origin and another one at 0.001. The high-pass frequency response is given by:

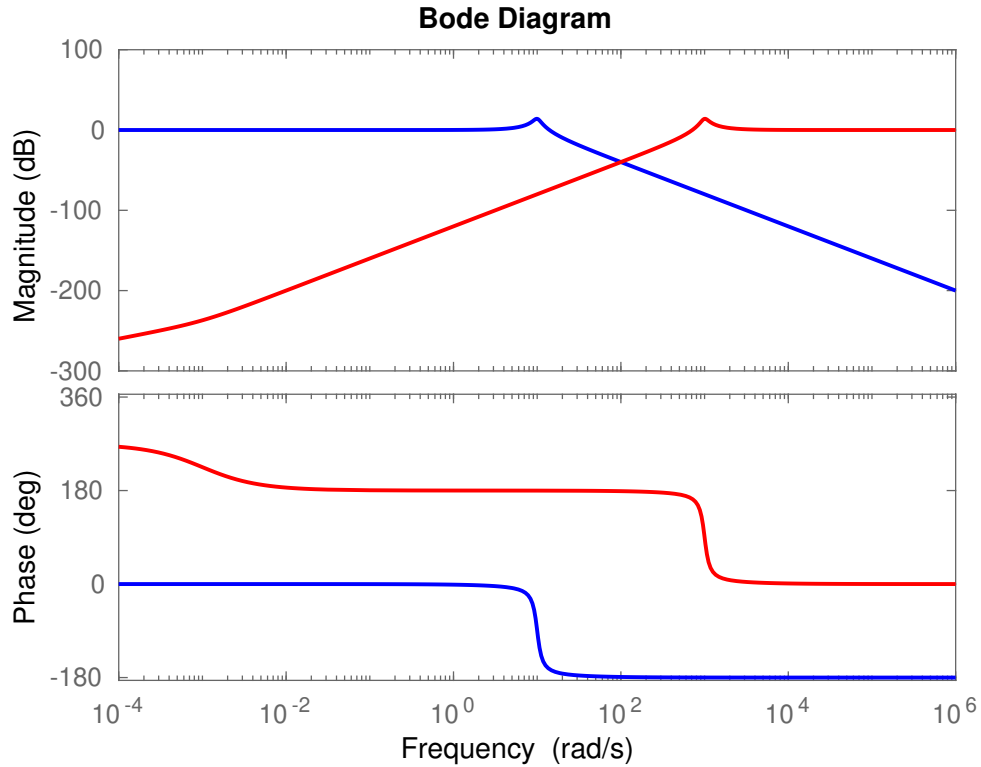$$G_{hp}(s) = \frac{s^2 - 0.001s}{s^2 + 200s + 10^6}. \tag{5.2}$$

Figure 5.4: Bode diagram of both modes. Blue represent the low-pass system (5.1) and red the high-pass system (5.2).

The resulting fourth order system is described as a state space representation, in a modal canonical form given by:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{5.3}$$

$$y(t) = Cx(t) + Du(t) \tag{5.4}$$

where $x(t) \in \mathbb{R}^4$ is the state vector, $u(t) \in \mathbb{R}^1$ is the single input vector and $y(t) \in \mathbb{R}^1$ is the single output vector, and

$$A = \begin{bmatrix} -100 & 994.99 & 0 & 0 \\ -994.99 & -100 & 0 & 0 \\ 0 & 0 & -1 & 9.949 \\ 0 & 0 & -9.949 & -1 \end{bmatrix}, \tag{5.5}$$

$$B = \begin{bmatrix} -24.6435 \\ -18.8943 \\ -4.1746 \\ -0.2675 \end{bmatrix}, \tag{5.6}$$

$$C = \begin{bmatrix} 24.41 & -21.2522 & -0.1537 & 2.3977 \end{bmatrix}, \tag{5.7}$$

$$D = 1. \tag{5.8}$$

Matrix A, given by (5.5), indicates two subsystem dynamically decoupled. The upper diagonal block refers to the high-pass system (5.2), whereas the lower diagonal block represents the low-pass system (5.1).

We simulate a pseudo-random binary sequence (PRBS) as input, with 200 samples, as shown in Figure 5.5. The PRBS signal is used as input to the low-pass and high-pass systems separately, considering two different time frames, of 2 and 20 seconds. The results of the four linear simulations are shown in Figure 5.6. The 20 seconds simulation for the high-pass system achieve steady state values very quickly, with a dynamics hard to be observed. Thus, we choose the 2 seconds time frame to simulate the fourth order serialized systems, given by (5.3)-(5.8). The result of the linear simulation of the fourth order system is presented in Figure 5.7.

System discretization is performed using Matlab™'s built-in function c2d to produce
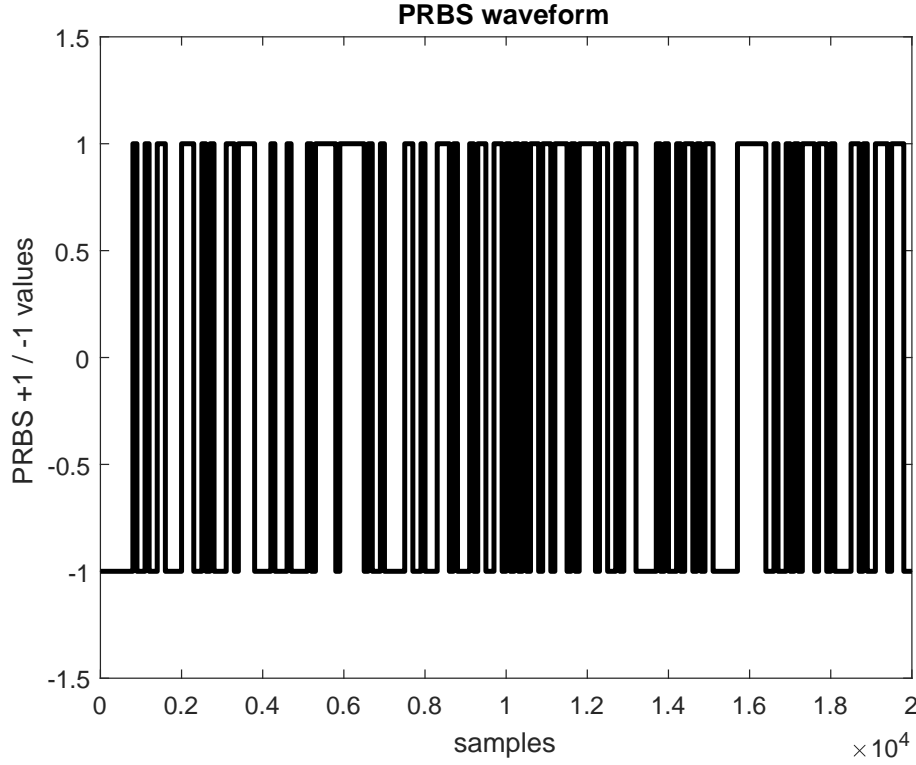
Figure 5.5: 200 PRBS samples used as input to simulate system (5.3)-(5.8)

a discrete-time state space representations according to

$$x(t_{k+1}) = A_d(t_k)x(t_k) + B_d u(t_k) + w(t_k), \tag{5.9}$$

$$y(t_k) = B_d x(t) + D_d u(t) + v(t_k) \tag{5.10}$$

where $t_{k+1} = t_k + h_k \in \mathbb{R}$ and $k \in \mathbb{N}$. $\rho(v(t_k)) = \mathcal{N}(0,R)$ and $\rho(w(t_k)) = \mathcal{N}(0,Q)$ are, respectively, the process and observation noise, with zero mean and covariance R and Q. When time-stamp is not available, the observation vector is approximated by $\tilde{y}_i \approx y(t_k)$, where i is the index of the next regular estimation time instant, multiple of T. When it is available, discretization is performed using variable time intervals.

5.3.2   Linear System

In this section we present one realization of the state estimation simulation for the system defined by (5.3)-(5.8), using the Kalman Filter described in Section 4.2.1 and the algorithms modifications explained in Section 4.3.
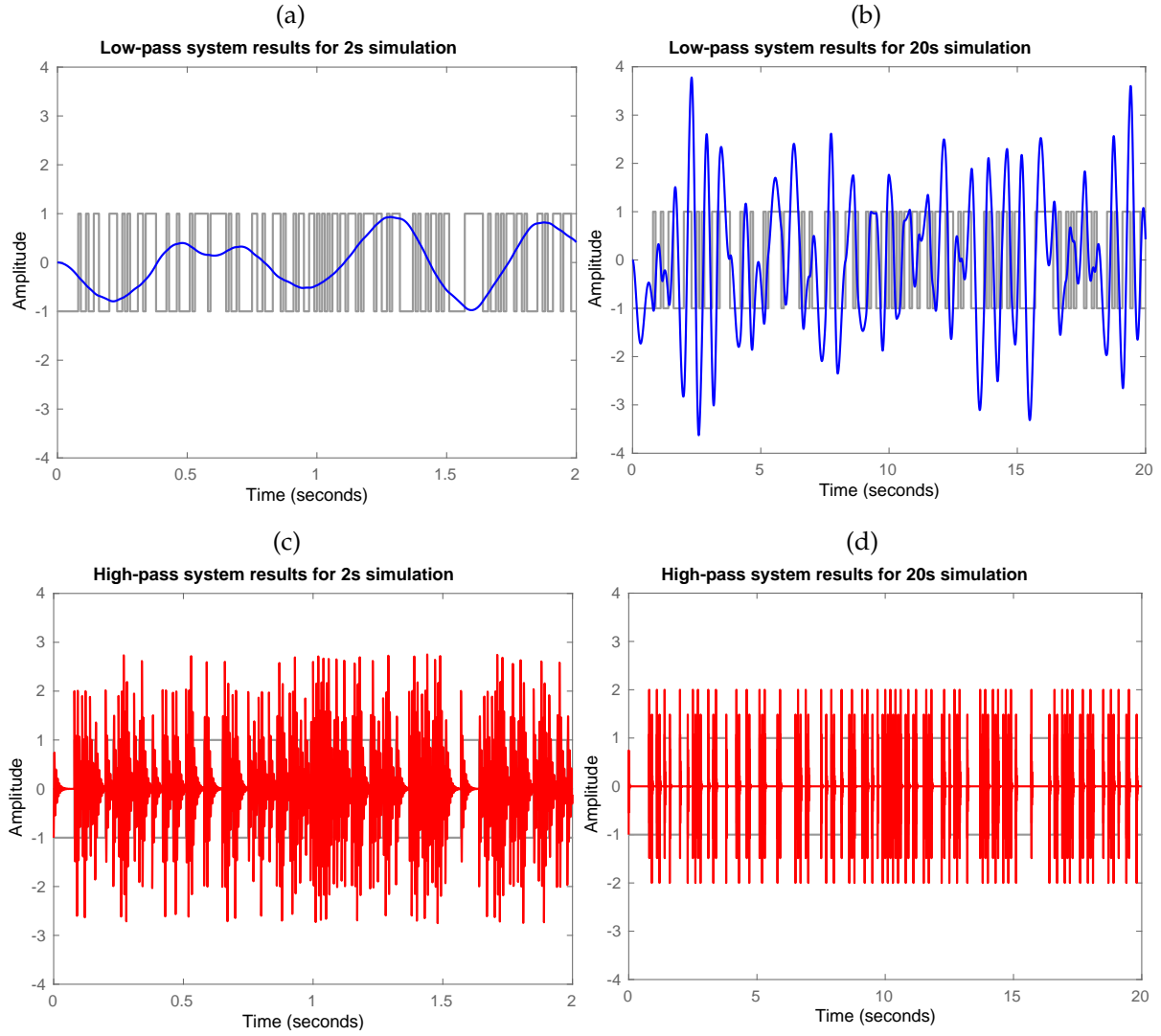
Figure 5.6: Linear simulation results of the low-pass and high-pass systems separately. Grey lines represent the PRBS input and colored lines the output. Graphs (a) and (b) with blue outputs represent the results of the low-pass system; and (c) and (d) with red outputs, the results of the high-pass system. Graphs on the left show a time frame of 2 seconds of simulations, whereas the ones on the right are the result of a 20 seconds time frame; considering the same 200 samples of PRBS for all four simulations.

Simulation parameters are set according to $\lambda = 500\,\mathrm{Hz}$, $SNR_{\mathrm{obs}} = 30\,\mathrm{dB}$, and $SNR_{\mathrm{pro}} = 30\,\mathrm{dB}$, where subscript obs and obs refer to observation and process, respectively. Time step to simulate nominal system model is set as $\delta t_{\mathrm{sim}} = 10^{-6}\,\mathrm{s}$ and the regular estimation time interval is given by $T = 2 \times 10^{-3}\,\mathrm{s}$. Noise is generated using the awgn function from Matlab™. For this realization, it resulted in a measurement noise variance of $R = 9.596 \times 10^{-4}$ and process noise variance of $Q = 9.986 \times 10^{-4}$. These
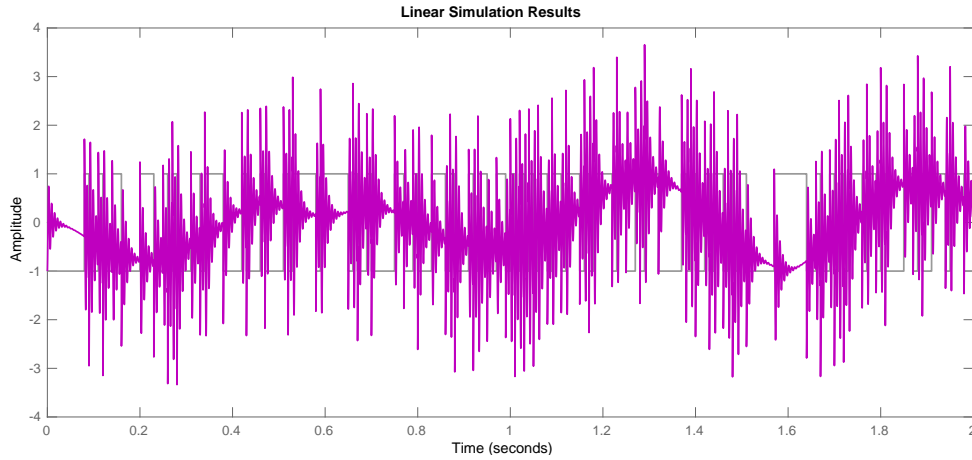
Figure 5.7: Output of the fourth order system to the PRBS input signal over a 2 seconds time frame.

exact values were used in the Kalman filter implementation and the estimation results for states x_1 and x_4 of the algorithms with and without timestamp, in comparison to the true state values are shown in Figures 5.8.

For the high-pass subsystem state x_1, it is hard to note a significant difference in the estimation performance. However, for the low-pass system state x_4, the degradation is visible for the algorithm that does not acknowledge time instants t_k information in the estimation process, represented by the dashed blue lines.

Figure 5.9 presents a data window from 0 to 0.013 seconds of the accuracy index RMSE of state x_2, for the algorithms with and without time-stamp. As expected, when first observation arrives at t_1, RMSE results distance themselves, in favor of the algorithm that considers time-stamp. The RMSE difference increases the next measurement arrives at t_2. For the time period without observations, indexes of both algorithms evolve in a similar way.

Finally, for this realization, we also present single-run consistency tests, using the indices NEES and NIS as defined in Section 4.4 in Figure 5.10. In each graph, the acceptance intervals are marked by horizontal lines. The upper plots (a) and (b) represent the NEES and NIS values, respectively, for the algorithm that considers time-stamp in the estimation process. We can see that the estimates are quite consistent, with the values inside the acceptance region most of the time. In fact, the null hypothesis rejection rate, that is the proportion of times the values were out of their acceptance region, was 4% for NEES, and 5,5% for NIS, very close the the 5% expected. When time-stamp information was not available, consistency was heavily degraded, with rejection
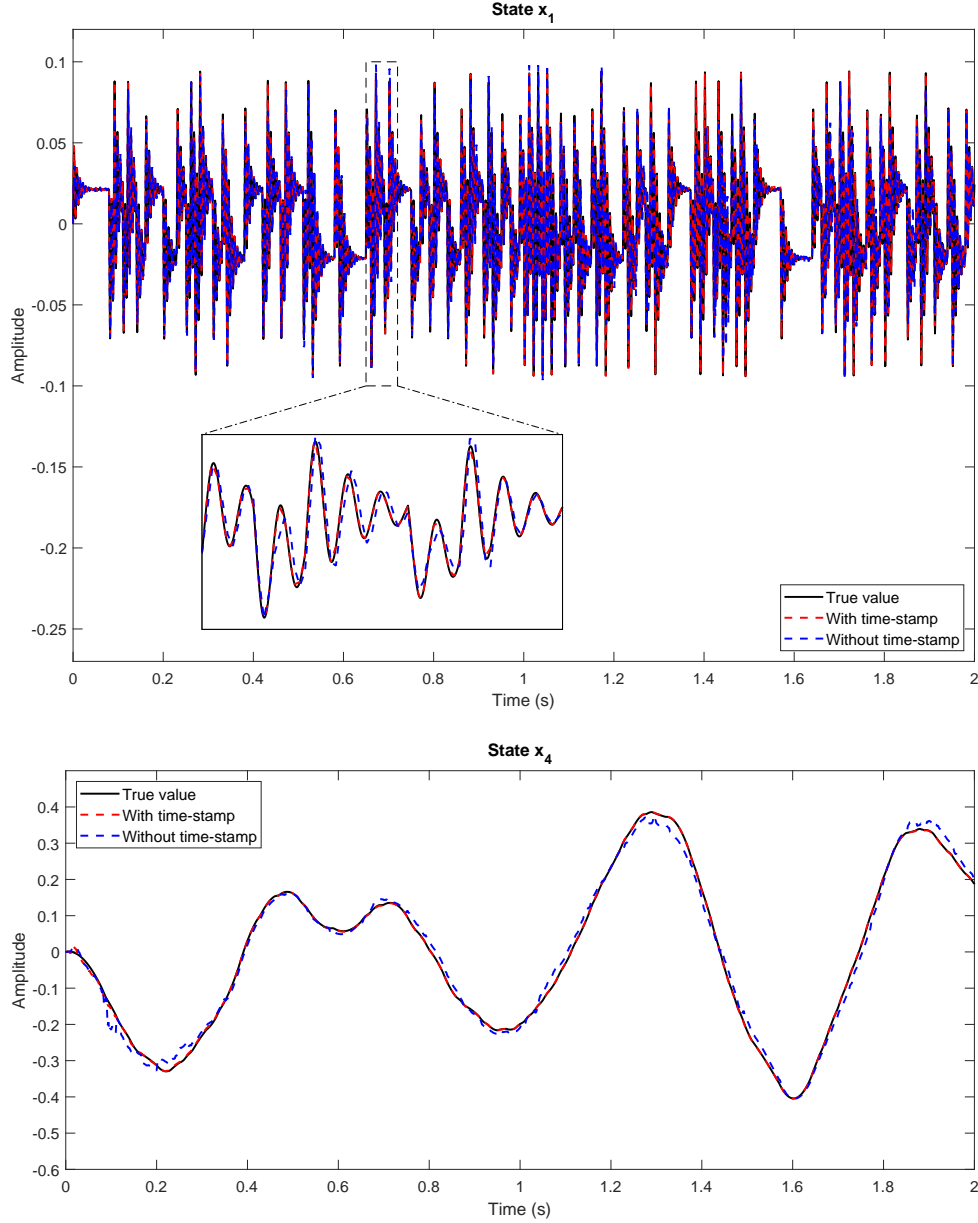
Figure 5.8: States x_1 and x_4 estimates with time-stamp (- -), without time-stamp (- -) and true values (—).

rates escalating to 61% and 43%, respectively. Moreover, NEES and NIS values were so high in some occasions, that the acceptance region is hardly visible in the graphs. In practice, such a consistency degradation means that estimation covariances are highly underestimated.

In the next sections we study the scenarios described in Section 5.2 for multiple runs. The simulation parameters are defined according to Table 5.2 Only the results of one state per mode are shown, namely x_1 and x_4, for simplicity sake.. Figure 5.11

Figure 5.9: Temporal cut from 0 to 0.013 seconds, for a realization of state x_2 RMSE for both estimators, with (—) and without (—) 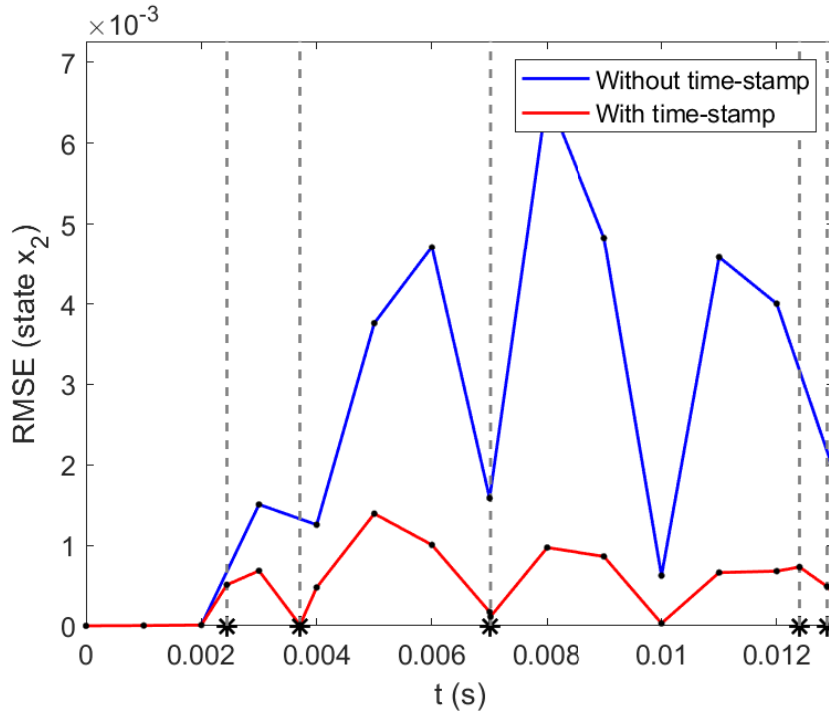time-stamp. Vertical dashed lines match the measurement sampling instants t_k. Black dots represent the estimation instants.

encompasses results for all scenarios. Each column contains the performance metrics variation for each scenario. A 100-run Monte Carlo simulation was performed for each of them, and the mean and 95% confidence intervals are presented for the algorithm with time-stamp in red and without time-stamp in blue.

Table 5.2: Linear system simulation parameters for SNR variation

Scenario	SNR (dB)	λ (Hz)	α
SNR variation	[60, 50, 40, 20, 10]	500	1
λ variation	30	[1000, 500, 200, 100]	1
α variation	30	500	[5, 3, 2, 1]

5.3.3 Measurement Signal-to-Noise Ratio Variation

Results from the first column of Figure 5.11 suggest that both agorithms with and without timestamp perform similarly for lower SNR in measurement and input data.
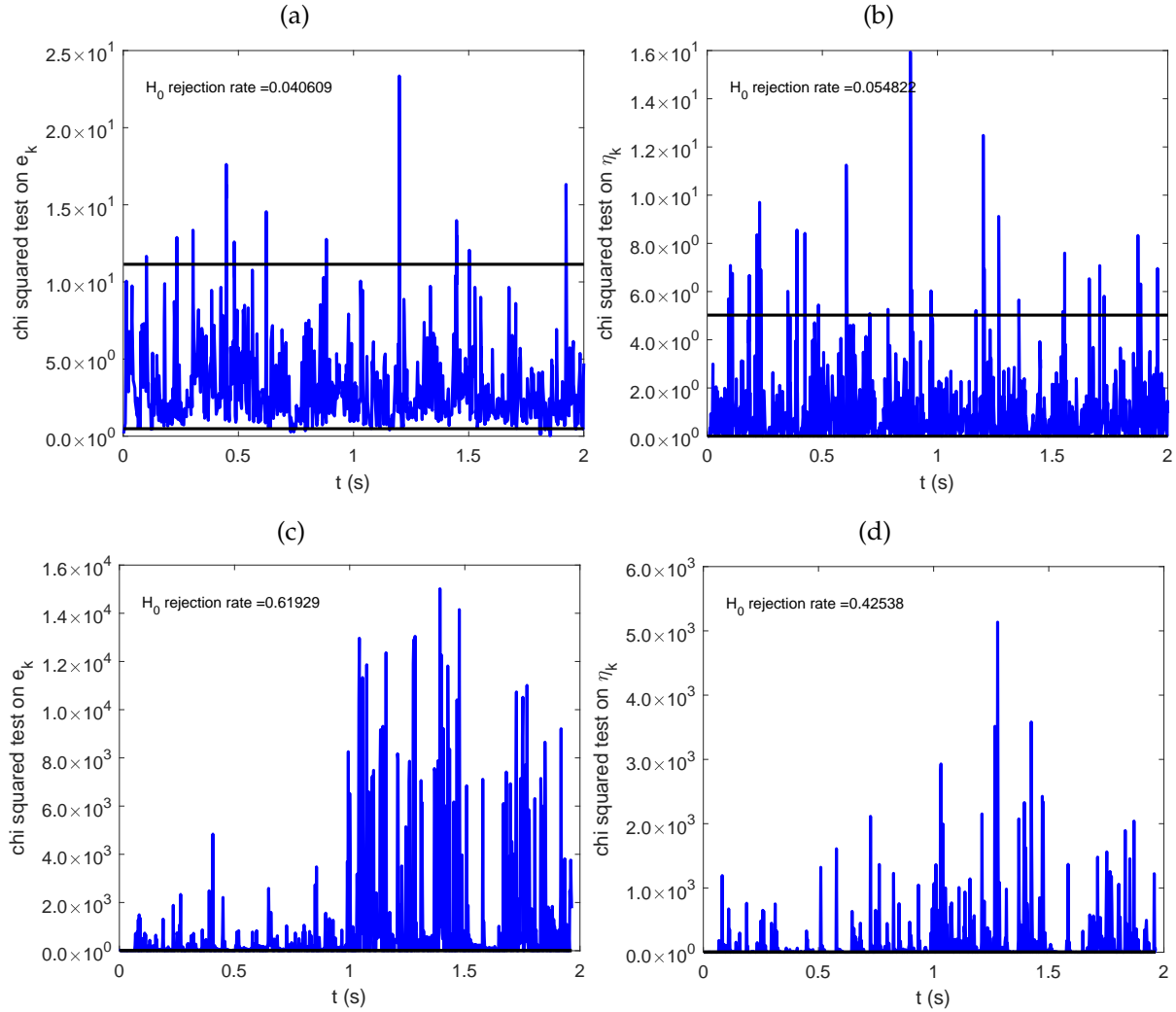
Figure 5.10: Consistency tests NEEs and NIS for the linear system state estimation with time-stamps (a) and (b); and without time-stamp (c) and (d). Horizontal lines define the acceptance region upper and lower limits, for a significance level $\alpha = 0.05$. In each graph, the null hypothesis H_0 rejection rate is also presented.

When there is lower noise levels, there are apparent differences in estimation performance in favor of considering timestamp.

There is a clear increasing trend for RMSE values of the algorithm considering timestamp, as expected, since state estimation with less noise in data should result in estimates closer to true values. As for consistency, considering time-stamp produces consistent estimates for all cases, with mean rates for NEES and NIS near the 5% significance level.

The same behavior cannot be observed for the method without timestamp. In fact, RMSE results reduce for higher noise levels in data. Such behavior can be explained by

the fact that for higher SNR, the additional noise tuned by assimilating data at incorrect time instants plays a more important role in the degradation of the estimates. When SNR decreases, the relevance of the additional noise caused by lack of timestamp is reduced. Estimates are more consistent and with less error in such scenarios.

5.3.4 Average Sampling Rate Variation

Variation of performance indices for λ variation are shown in the second column of Figure 5.11. RMSE results for the high-pass system have such a high variability that it is hard to take any conclusions on the behavior. Mean RMSE for the algorithm with timestamp is systematically slower, however, and RMSE values are quite small, specially in comparison to results from the low-pass system. Graph (e), on the other hand, showing RMSE for low-pass system state x_4, suggest that for smaller average frequencies of the aperiodic sampled observations, not considering the exact time they were taken degrades accuracy greatly. That is expected, since the approximation errors of $\tilde{y}_i \approx y(t_k)$ are higher for more sparse average time intervals $1/\lambda$ and so is the additional noise introduced by data assimilation performed at incorrect time instants. We can also observe a small degradation for the algorithm with timestamp, which can be explained by discretization errors, increased for larger time steps.

Consistency performance results show that, for the parameter combinations simulated, not considering timestamp always underestimated errors, while the algorithm with timestamp achieved rejection rates consistently closer to 5%.

5.3.5 Regular and Average Irregular Time Interval Relation Variation

Similarly to the variation of λ, changing values of α resulted in high variability in RMSE values for state x_1 with very small means. It is possible to observe an increasing trend of RMSE for the algorithm without timestamp, for smaller α values. The same trend is much more apparent in RMSE for low-pass system state x_1. On the other hand, state estimate errors for the algorithm with timestamp are small and apparently constant in all cases. It suggests that, the more sparse are the observation time intervals in comparison to the regular estimation time interval, that is the higher the α, the more important it is assimilate data at the correct time instants t_k.

Again, consistency test results showed underestimated errors for the algorithm without timestamp systematically, whereas considering timestamp produced much more consistent estimates.
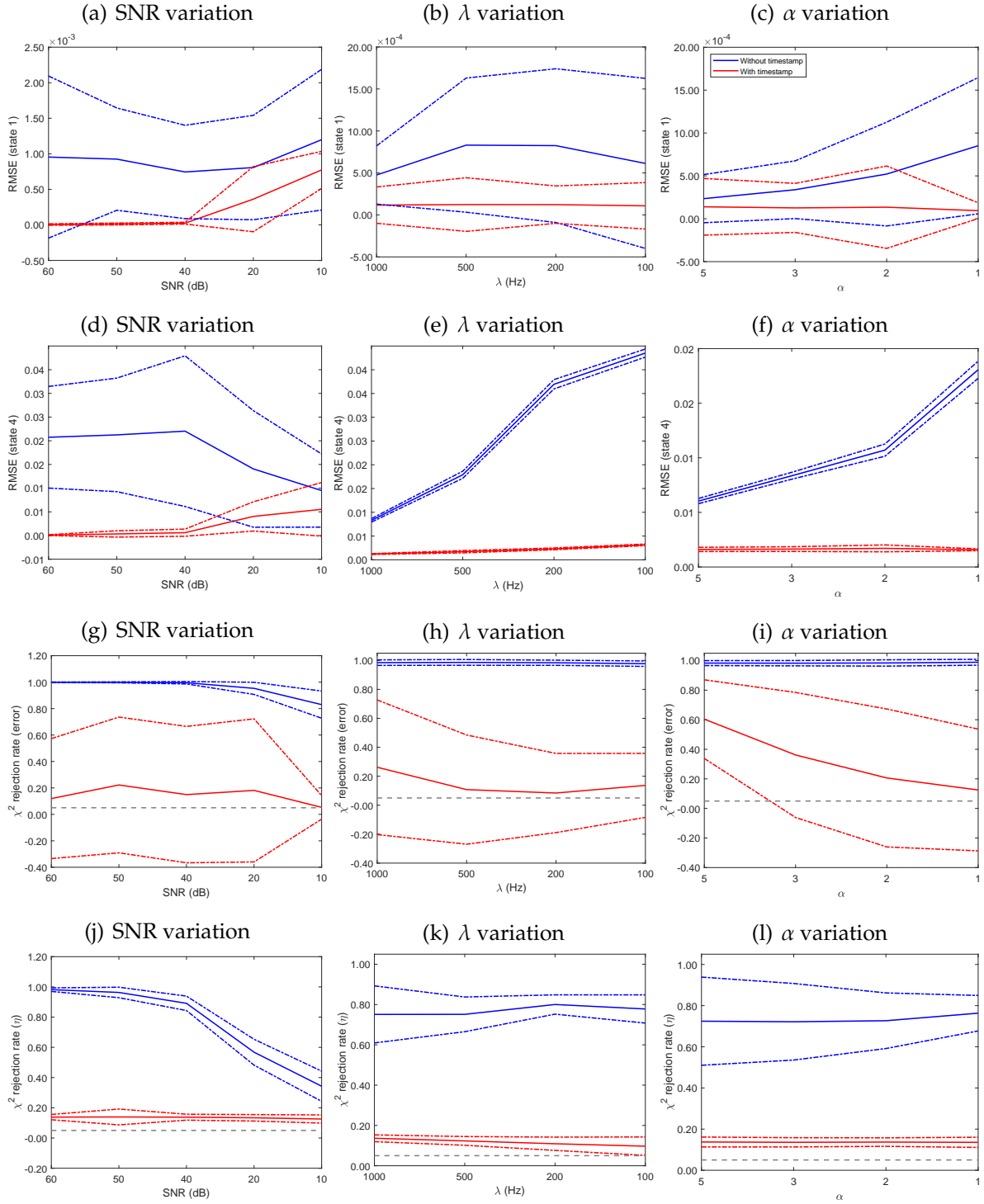
Figure 5.11: Linear system estimation mean performance indices with 95% confidence intervals, as a function of SNR (a,d,g,j), λ (b,e,h,k) and α (c,f,i,l), considering (—) and not considering (—) time-stamp. RMSE results are shown in the first (x_1) and second (x_4) rows. Consistency results are in third (NEES)) and fourth (NIS) rows. Horizontal gray dashed lines represent the 5% significance level for the consistency tests.

5.4 Nonlinear System: Unicycle Position

We now consider a nonlinear system given by a nonholomonic moving unicycle robot, whose position in the xy-plane must be estimated. In many real target tracking problems such as this, measurements are available by unsynchronized cameras through an internet network, for instance. In other applications multiple GPS sensors might be used in a centralized architecture without data alignment or association. These configurations are susceptible to sampling irregularities, which motivates our choice.

5.4.1 System Description

Consider a nonholomonic moving robot, with the cinematic process model given by

$$
\begin{aligned}
\dot{p}_x &= v\cos(\theta), \\
\dot{p}_y &= v\sin(\theta), \\
\dot{\theta} &= u_1(t), \\
\dot{v} &= u_2(t),
\end{aligned}
\tag{5.11}
$$

where p_x and p_y are the position coordinates, θ the angular orientation, v the linear velocity and inputs u_1 and u_2 are the angular velocity ω and the linear acceleration a, respectively. Figure 5.12 shows a schematic of the robot and its states,
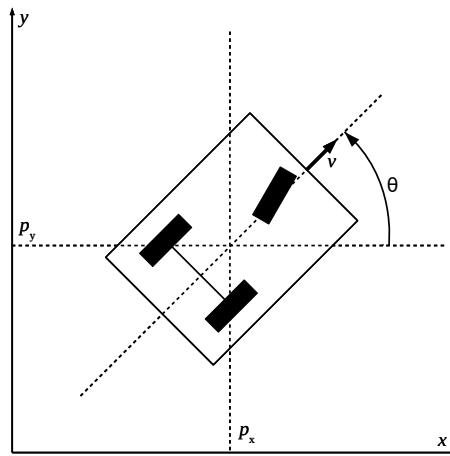

Figure 5.12: Nonholomonic robot system representation. The system states p_x, p_y, v and θ are highlighted.

The system described by (5.11) is discretized by a fourth order Runge-Kutta method as described in Section 4.1 and the discrete state vector x_i is given by $x_i \triangleq [p_{x,i}\ p_{y,i}\ \theta_i\ v_i]^T$, where subscript i indicate discrete instants iT.

The observation model $y(t_k) \in \mathbb{R}^2$

$$y(t_k) = \begin{bmatrix} p_x(t_k) \\ p_y(t_k) \end{bmatrix} + v(t_k), \tag{5.12}$$

is given by the position coordinates, and $\rho(v(t_k)) = \mathcal{N}(0, R_{t_k})$ is the observation noise PDF, with zero mean and covariance R_{t_k}. Discrete time instants with t_k are determined by random time intervals h_k, that is $t_{k+1} = t_k + h_k$. When time-stamp information about t_k is not available, the observation vector is approximated by $\tilde{y}_i \approx y(t_k)$, where i is the index of the next time instant, multiple of T.

Input vector $u_i = [\omega_i\ a_i]^T$ is measured by a girometer and accelerometer, respectively. We assume that

$$u_i = \tilde{u}_i - w_i, \tag{5.13}$$

where $\tilde{u}$ represent the sensors' measured values and $\rho(w_i) = \mathcal{N}(0, Q_i)$ is the corresponding noise PDF, with zero mean and covariance Q_i.

We simulate 60 seconds of robot movement, considering a step size of $\delta t_{\text{sim}} = 10^{-4}$. Input signals are generated arbitrarily according to Figure 5.13.

Accuracy metrics for the unicycle position system is a modification of the RMSE defined in (4.120), to reflect error in xy-plane position, given by:

$$RMSE_{\text{position}} = \frac{\sum_{i=1}^{N} \sqrt{(\hat{p}_{x,i} - p_{x,i})^2 + (\hat{p}_{y,i} - p_{y,i})^2}}{N}, \tag{5.14}$$

where $\hat{p}_{x,i}$ and $\hat{p}_{y,i}$ are position estimates at regularly spaced time intervals T, $p_{x,i}$ and $p_{y,i}$ are true position coordinates of the unicycle, at the same time instants and N is the total amount of estimates.

5.4.2   Unicycle Position System

Figure 5.14 shows the robot trajectory on the xy-plane for the given input signals, leaving from point (0,0), as well as a realization of noisy and aperiodic measurements

as red dots, with signal-to-noise ratio of $SNR_y = 30$ dB and $\lambda = 3.33$ Hz. Dashed blue line represents UKF estimation, considering time-stamp, $\alpha = 5$ and $SNR_u = 10$ dB.
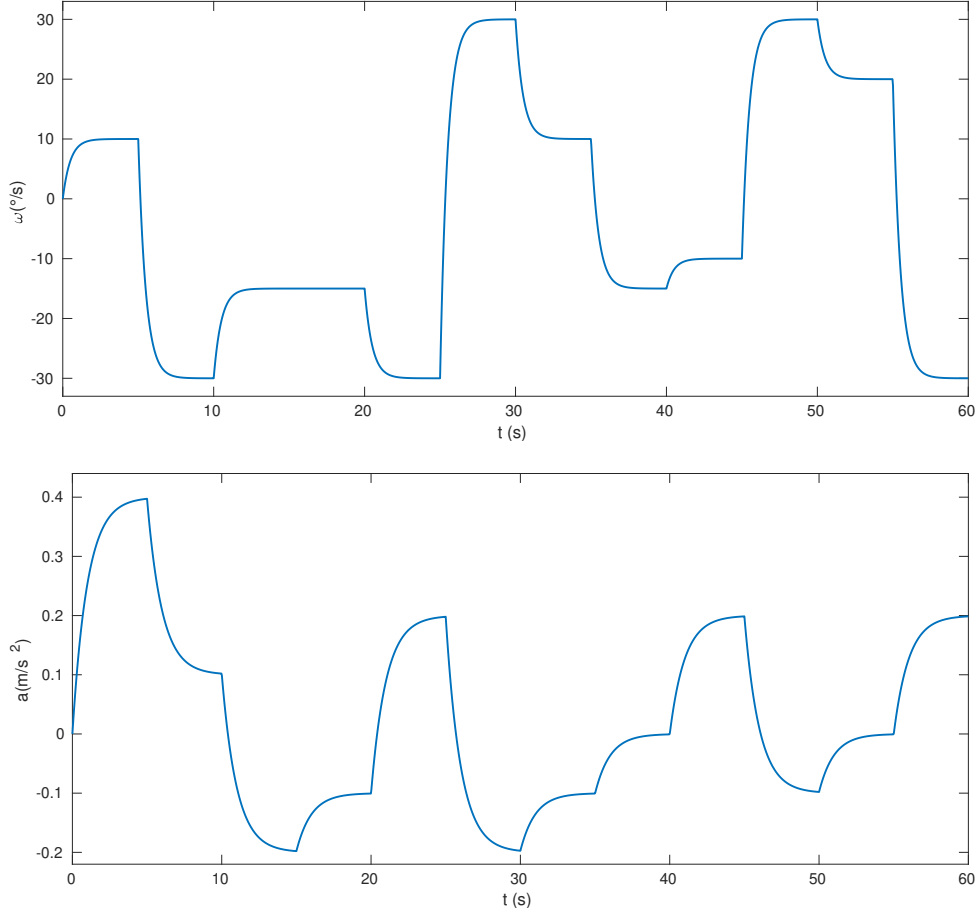
Figure 5.13: Simulation input signals. (a) shows temporal sequence of angular velocity ω and (b) shows linear acceleration a.

Figure 5.15 shows a timespan from 0 to 1.5 seconds of a realization of position RMSE, considering $\lambda = 0.5$, $\alpha = 2$, $SNR_y = 40$ dB and $SNR_u = 20$ dB, for the UKF considering and not considering time-stamp. Black dots represent the regular time instants kT whereas the asterisks on x-axis and dashed vertical lines match the exact measurement instants t_k. The first two measurements, before 0.25 seconds, are very close to the first regular time instant $1T = 0.25$ s and thus cause no significant difference on performance. It is expected, since the approximation error due to $\tilde{y}_i \approx y(t_k)$ is irrelevant. Next measurement t_3, taking place around 0.8 seconds, when assimilated correctly reduces the estimate error at the correct instant. When assimilated at the next estimation time instant, the position error is not decreased appropriately. At t_4, around

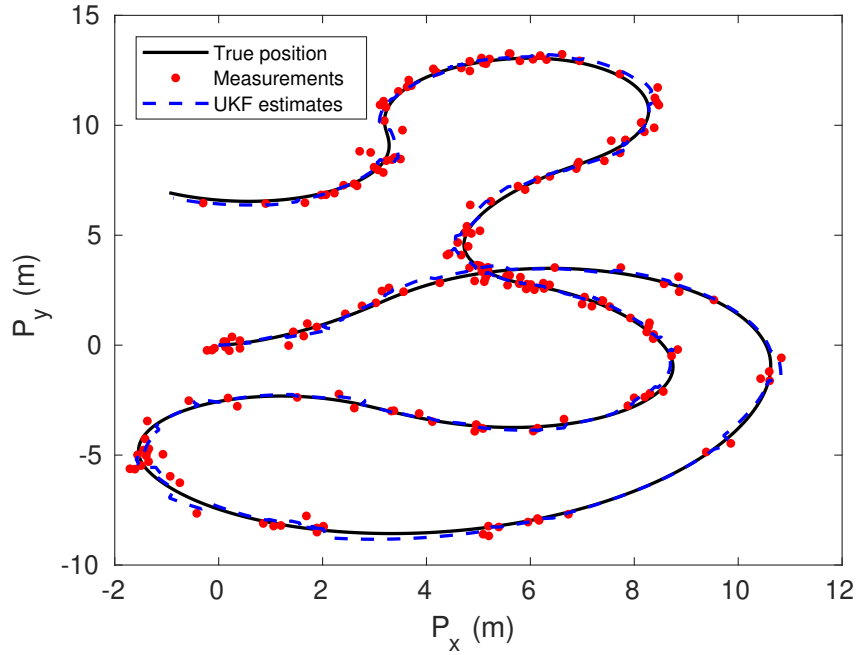Figure 5.14: True position (—), noisy measurements (·) and UKF estimates (- -) considering time-stamp, for a measurement noise of $SNR_y = 30$ dB, $\lambda = 0.3$ s and $\alpha = 5$.

1.1 seconds, we can see another effect of assimilating the measurement at the correct time, with the red line decreasing slightly.

In the next sections we perform Monte Carlo simulations, with 1000 runs per combination of parameters, for the scenarios from Section 5.2. Simulation parameters used for the nonlinear system are shown in Table 5.3. For all cases, input SNR is constant at 20 dB. All results are comprised in Figure 5.16, with the evolution of mean values in solid lines and the corresponding 95% confidence interval in dashed lines. First row of graphs shows accuracy results for position RMSE, as described in (5.14). Second and third rows present the consistency results for NEES and NIS tests, respectively, as described in Section 4.4. The blue lines are for the algorithm that considers time-stamp and red lines are for the one without time-stamp.

5.4.3   Measurement Signal-to-Noise Ratio Variation

Results from the first column of Figure 5.16 show the variation in performance indices for the different measurement SNR levels simulated. Both algorithms have their estimate error increased, when measurement data is contaminated with higher noise levels, as expected. It is interesting to observe, however, that the mean values
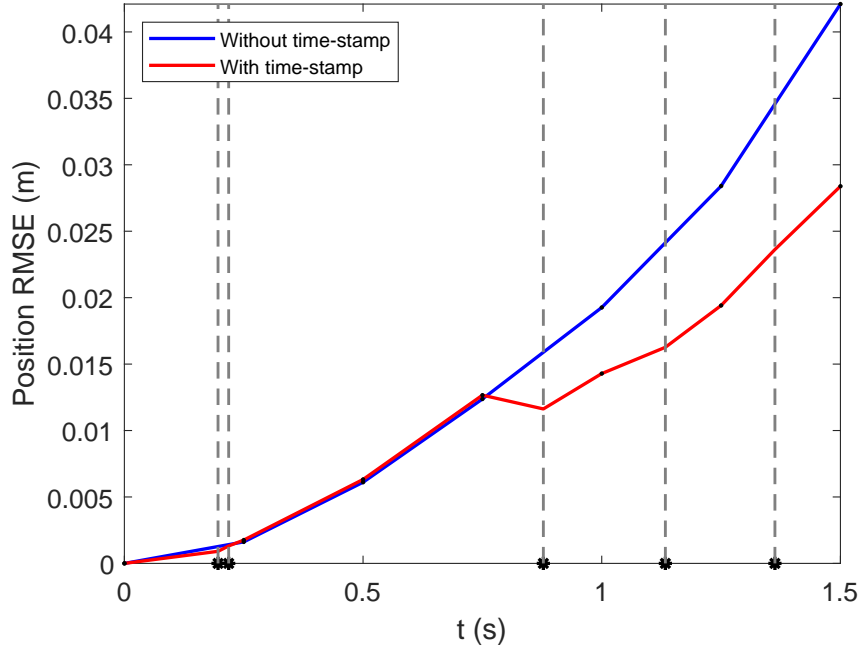
Figure 5.15: Temporal cut from 0 to 1.5 seconds, for a realization of J of both estimators, with and without time-stamp. Asterisks on x axis and vertical dashed lines match the measurement sampling instants t_k. Black dots represent the regular estimation instants, same as input regular sampling instants.

Table 5.3: Unicycle position estimation simulation parameters for observation SNR variation

Scenario	SNR observations (dB)	λ (Hz)	α
SNR variation	$[100, 80, 60, 40, 20, 10]$	10	2
λ variation	40	$[10, 5, 0.33, 2.5, 2, 1.67]$	2
α variation	40	10	$[10, 5, 2, 1]$

are clearly different when there is less noise, in favor of considering timestamp in the estimation process. As SNR is decreased to 20 and 10 dB, however, both RMSE have no statistical difference. It can be explained similarly to what was observed in the linear system. The higher the noise, the more relevant are the approximation errors of $\tilde{y}_i \approx y(t_k)$. When measurements are highly contaminated with noise, these approximation errors get irrelevant for the estimation process, and both algorithms perform similarly.

Same behavior can be observed in the consistency tests results. Estimates are inconsistent for the algorithm without timestamp only for higher SNR. When noise

levels increase, both algorithms have similar consistency results. Graph (g) shows a slight inconsistency in the estimation of the algorithm with timestamp, for 100 db SNR. It is probably due to numerical errors caused by very small numbers for measurement covariances, when there is virtually no noise in data.

5.4.4   Average Sampling Rate Variation

As for different λ values, variation of indices is presented in the second column of Figure 5.16. RMSE results suggest a slightly higher performance degradation for more sparse time intervals, providing that dynamics and other parameters are kept constant. Estimate errors seem to increase faster for the algorithm without timestamp, due to higher approximation errors of $\tilde{y}(i) \approx y(t_k)$. Another source of accuracy deterioration is the discretization error, higher for more sparse time intervals.

NEES and NIS test results suggest that both algorithms produce estimates with similar consistency levels. The reason why higher differences are not observed may derived from the use of UKF, that does not propagate error covariances, but construct them at every estimation step.

5.4.5   Regular and Average Irregular Time Interval Relation Variation

Finally, the last scenario results are shown in the third column of Figure 5.16. They suggest that estimation performance difference is only significant for higher α values. Both consistency and accuracy results are similar for $\alpha = [2,1]$, when estimation is performed considering and not considering timestamp.

For the accuracy metric, we observe an opposite trend for the nonlinear system in comparison to the linear system. Whereas estimate errors increased for higher α values in the linear case, here they decrease. We note that the former simulation considered SNR of 30 dB for both measurements and input signals. In this case, however, SNR for measurements are considered to be higher than for input data, 40 dB and 20 dB, respectively. Therefore, the nonlinear state estimation has a higher confidence in the observation model. And, since reducing α values result in less sparse time intervals of observations, in comparison to input time interval, the data assimilation step is performed more frequently and increases estimation accuracy.

We also observe that the results from the algorithm with timestamp remained approximately constant throughout all cases. It indicates that when exact time instants

t_k are available to the estimator, the ratio given by α has no impact in performance. If they are not available, however, estimate errors can be multiplied by 4, if α varies from 1 to 10.
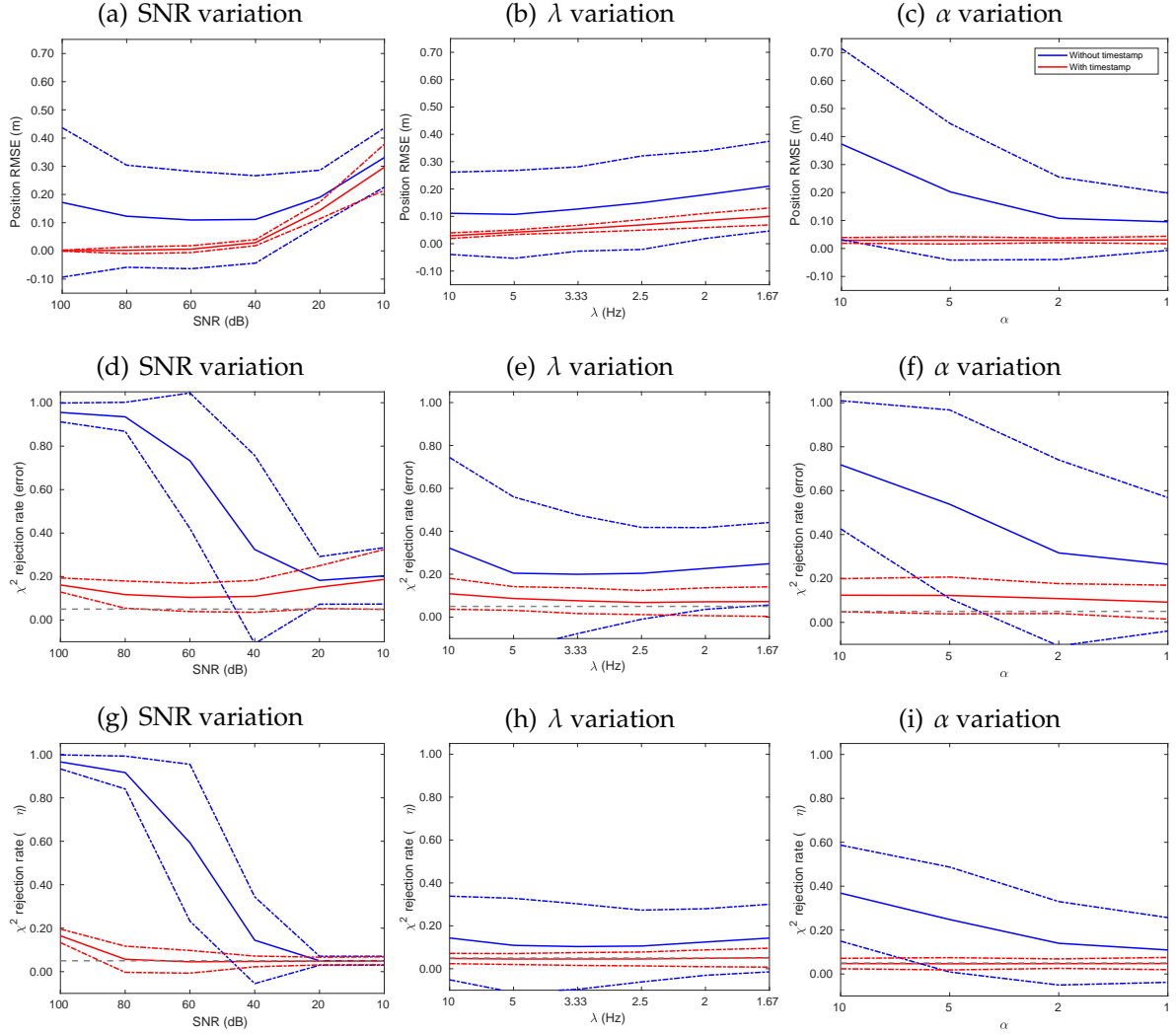
Figure 5.16: Nonlinear system estimation performance indices with 95% confidence intervals, as a function of SNR (a,d,g), λ (b,e,h) and α (c,f,i), for algorithms with (——) and without (——) time-stamp. Position RMSE results are shown in the first row. Consistency results are in second (NEES)) and third (NIS) rows. Horizontal gray dashed lines represent the 5% significance level for the consistency tests

5.5   Simulated Results Discussion

Two different system are simulated to address the effects of neglecting timestamp in state estimation performance: a linear fourth order system, with two underdamped

modes; and a nonlinear unicycle position system. We design three scenarios in which only one parameter of interest is changed per scenario, while the others are kept constant. That way, we can infer the contribution to performance variation to each of the parameter: SNR; λ; and α.

For both systems, performance metrics obtained by neglecting time-stamp had a higher variability in most of the simulations, which is by itself a drawback of not considering timestamp. It indicates that that estimate accuracy and consistency are less reliable in such cases.

The average performance metrics for the algorithm considering time-stamp were always higher or equal to the metrics from the algorithm without time-stamp, as expected. Accuracy performance for SNR variation suggested that for lower noise levels or higher SNR, the relevancy of considering or neglecting time-stamp is much higher. As noise levels increase, performance for both algorithms tend to approximate. In both cases, the lowest SNR simulated produced RMSE results that are statistically indistinguishable.

For different λ values, the algorithm that considers time-stamp maintained its performance metrics almost constant, specially for consistency tests, as it was expected. Results from the algorithm without timestamp indicates that more sparse observation time intervals deteriorate estimation performance, due to higher approximation errors. This deterioration is less significant for systems with faster dynamics, such is the case of the high-pass system, whose state estimate errors were similar for both algorithms, for different λ values.

Finally, results from of performance behavior for the variation of α values show different behaviors for the linear and nonlinear system for the algorithm without timestamp. The linear case indicate an increase in state estimates RMSE, for smaller α values. Simulated parameters consider same SNR for both input and measurement data. So the more sparse are the observations, the higher are the approximation errors of assimilating data at incorrect time instants t_k, given by $\tilde{y}(i) \approx y(t_k)$. In the nonlinear case, we consider less noise in measurement, with two times higher SNR, that is $SNR_{\text{observation}} = 40$ dB and $SNR_{\text{input}} = 20$. Therefore, although approximation errors of $\tilde{y}(i) \approx y(t_k)$ increase for more sparse time intervals in the observations, data assimilation steps happen more frequently, contributing to a reduction of estimate errors.

Conclusions

Falta conclusão

6.1 Project Overview

Investigou-se isso e aquilo... (caso algum leitor pule direto para a conclusão)

6.2 Main Results

Como os resultados corroboram os objetivos perseguidos. O que é relevante dizer sobre cada objetivo descrito no capítulo 1, com o que foi feito no trabalho.

Autocrítica sobre o trabalho: alguma hipótese foi pouco realista? o que não foi feito tão bem? o que foi feito bem?

6.3 Future Work

Talvez uma lista itemizada Trabalhos futuros podem vir diretamente da autocrítica.

Bibliography

Aguirre, L. A., Teixeira, B. O. S., and Tôrres, L. A. B. (2005). Using data-driven discrete-time models and the unscented Kalman filter to estimate unobserved variables of nonlinear systems. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 72(2):1–12.

Albertos, P., Peñarrocha, I., and Garcia, P. (2004). Virtual sensors under delayed scarce measurements. *7th IFAC Symposium on Cost Oriented Automation*, 37(5):85–90.

Andler, S. F. and Brohede, M. (2009). Information Fusion from Databases, Sensors and Simulations: Annual Report 2008. *Skövde: Institutionen för kommunikation och information*, page 65.

Anxi, Y., Diannong, L., Weidong, H., and Zhen, D. (2005). A unified out-of-sequence measurements filter. In *IEEE National Radar Conference - Proceedings*, pages 453–458.

Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A Tutorial on Particle Filters for Online Nonlinear / Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188.

Axelsson, P. and Gustafsson, F. (2015). Discrete-time solutions to the continuous-time differential Lyapunov equation with applications to Kalman filtering. *IEEE Transactions on Automatic Control*, 60(3):632–643.

Bar-Shalom, Y. (2000). Update with out-of-sequence measurements in tracking: Exact solution. *Proceedings of SPIE*, 4048(3):769–778.

Bar-shalom, Y., Daum, F., and Huang, J. I. M. (2009). The Probabilistic Data Association Filter. *IEEE Control Systems Magazine*, 29(6):82–100.

Bar-Shalom, Y., Rong Li, X., and Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, Inc.

Bergman, N. (1999). *Recursive Bayesian Estimation - Navigation and Tracking Applicationss*. PhD thesis, Linköping University.

Boström, H., Andler, S. F., Brohede, M., Johansson, R., Karlsson, E., Laere, J. V., Niklasson, L., Nilsson, M., Persson, A., and Ziemke, T. (2007). On the definition of information fusion as a field of research. Technical report, University of Skösvde.

Brahmi, M., Schueler, K., Bouzouraa, S., Maurer, M., Siedersberger, K. H., and Hofmann, U. (2013). Timestamping and latency analysis for multi-sensor perception systems. *Proceedings of IEEE Sensors*.

Castanedo, F. (2013). A review of data fusion techniques. *The Scientific World Journal*, 2013.

Challa, S., Evans, R. J., and Wang, X. (2003). A Bayesian solution and its approximations to out-of-sequence measurement problems. *Information Fusion*, 4:185–199.

Chen, B., Yu, L., Zhang, W.-A., and Liu, A. (2013). Robust Information Fusion Estimator for Multiple Delay-Tolerant Sensors With Different Failure Rates. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(2):401–414.

Chen, C.-T. (1999). *Linear System Theory and Design*. Oxford University Press.

Dasarathy, B. (2001). Information Fusion - what, where, why, when, and how? *Information Fusion*, 2(2):75–76.

Dasarathy, B. V. (1997). Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proceedings of the IEEE*, 85(1):24–38.

Dasarathy, B. V. (2000). More the merrier...or is it? - Sensor suite augmentation benefits assessment. *Proceedings of the 3rd International Conference on Information Fusion, FUSION 2000*, 2:20–25.

Dubois, D. and Prade, H. (2000). Possibility theory in information fusion. *Proceedings of the 3rd International Conference on Information Fusion, FUSION 2000*, 1.

Durrant-Whyte, H. F. (1988). Sensor Models and Multisensor Integration. *The International Journal of Robotics Research*, 7(6):97–113.

Elmenreich, W. (2002). An Introduction to Sensor Fusion. Research report, Vienna University of Technology.

Elson, J., Girod, L., and Estrin, D. (2002). Fine-grained network time synchronization using reference broadcasts. *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI)*, 36(SI):147.

Eng, F. and Gustafsson, F. (2005). System Identification Using Measurements Subject to Stochastic Time Jitter. *IFAC Proceedings Volumes*, 38:1179–1184.

Faragher, R. (2012). Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]. *IEEE Signal Processing Magazine*, 29(5):128–132.

Fatehi, A. and Huang, B. (2017). Kalman filtering approach to multi-rate information fusion in the presence of irregular sampling rate and variable measurement delay. *Journal of Process Control*, 53:15–25.

Foster, J. L. and Hall, D. K. (1981). Muitisensor Analysis of Hydrologic Features with Emphasis on the. *Photogrammetric Engineering and Remote Sensing*, 47(5):655–664.

Fowler, C. A. (1979). Comments on the Cost and Performance of Military Systems. *IEEE Transactions on Aerospace and Electronic Systems*, AES-15(1):2–10.

Fridman, E. (2014). Introduction to time-delay and sampled-data systems. In *2014 European Control Conference (ECC)*, pages 1428–1433. IEEE.

Ganeriwal, S., Kumar, R., and Srivastava, M. B. (2003). Timing-sync protocol for sensor networks. In *Proceedings of the first international conference on Embedded networked sensor systems - SenSys '03*, page 138.

Goodman, I. R., Mahler, R. P., and Nguyen, H. T. (1997). *Mathematics of Data Fusion*. Kluwer Academic Publishers.

Gopalakrishnan, A., Kaisare, N. S., and Narasimhan, S. (2010). Incorporating delayed and infrequent measurements in Extended Kalman Filter based nonlinear state estimation. *Journal of Process Control*, 21:119–129.

Gravina, R., Alinia, P., Ghasemzadeh, H., and Fortino, G. (2017). Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges. *Information Fusion*, 35:68–80.

Hadidi, M. and Schwartz, S. (1979). Linear recursive state estimators under uncertain observations. *IEEE Transactions on Automatic Control*, 24(6):944–948.

Hall, D. L. and Llinas, J. (1997). An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23.

Han, C. and Zhang, H. (2009). Linear optimal filtering for discrete-time systems with random jump delays. *Signal Processing*, 89(6):1121–1128.

He, L. M. (2008). Time synchronization based on spanning tree for wireless sensor networks. *2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008*, pages 4–7.

Hu, R. J., Wang, Z., Alsaadi, F. E., and Hayat, T. (2017). Event-based filtering for time-varying nonlinear systems subject to multiple missing measurements with uncertain missing probabilities. *Information Fusion*, 38:74–83.

Huck, T., Westenberger, A., Fritzsche, M., Schwarz, T., and Dietmayer, K. (2011). Precise timestamping and temporal synchronization in multi-sensor fusion. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 242–247. IEEE.

Hyberts, S. G., Robson, S. A., and Wagner, G. (2013). Exploring signal-to-noise ratio and sensitivity in non-uniformly sampled multi-dimensional NMR spectra. *Journal of Biomolecular NMR*, pages 167–178.

Jaffer, A. G. and Gupta, S. C. (1971). Recursive Bayesian Estimation With Uncertain Observation. *IEEE Transactions on Information Theory*, I(September):614–616.

Jazwinski, A. (1970). *Stochastic Processes and Filtering Theory, Volume 64*. Academic Press.

Jing, Z., Pan, H., and Qin, Y. (2013). Current progress of information fusion in China. *Chinese Science Bulletin*, 58(36):4533–4540.

Jordao, A., Torres, L. A. B., and Schwartz, W. R. (2018). Novel approaches to human activity recognition based on accelerometer data. *Signal, Image and Video Processing*, 12(7):1387–1394.

Julier, S. J. and Uhlmann, J. K. (1997). A New Extension of the Kalman Filter to Nonlinear Systems. *Proc. AeroSense: 11th Int. Symp. Aerospace/Defence Sensing, Simulation and Controls.*

Julier, S. J. and Uhlmann, J. K. (2004). Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, 92(3):401–422.

Julier, S. J. and Uhlmann, J. K. (2005). Fusion of time delayed measurements with uncertain time delays. In *Proceedings of the 2005 American Control Conference*, pages 4028–4033.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of ASME – Journal of Basic Engineering*, 82:35–45.

Kanchanaharuthai, A. and Wongsaisuwan, M. (2002). Stochastic H/sub 2/-optimal controller design for sampled-data systems with random sampled measurement. In *Proceedings of the 41st SICE Annual Conference. SICE 2002.*, volume 3, pages 2042–2047. Soc. Instrument & Control Eng. (SICE).

Kaur, P. and Abhilasha (2015). Time synchronization in wireless sensor networks: a review. *International Journal of Computer Engineering and Applications*, IX(VI):403–414.

Khaleghi, B., Khamis, A., Karray, F. O., and Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44.

Kolmogorov, A. N. (1962). *Interpolation and extrapolation of stationary random sequences.* Translated by W. L. Doyle, and Ivan Selin, RAND Corporation.

Kordestani, M. and Saif, M. (2017). Data Fusion for Fault Diagnosis in Smart Grid Power Systems. In *IEEE 30th Canadian Conference on Electrical and Computer Engineering*, pages 516–521.

Kunoh, F. M. (2015). *Procedimento para Detecção de Falseamento via Amostragem Não Uniforme.* PhD thesis, Universidade Federal de Minas Gerais.

Kwok, C., Fox, D., and A, M. M. (2004). Real-Time Particle Filters. *Proceedings of the IEEE*, 92(3):469 – 484.

Leadbetter, M. R., Lindgren, G., and Rootzén, H. (1983). *Extremes and Related Properties of Random Sequences and Processes.* Springer-Verlag.

Lin, H. and Sun, S. (2016). Distributed fusion estimator for multi-sensor asynchronous sampling systems with missing measurements. *IET Signal Processing*, 10(7):724–731.

Liu, Q., Wang, Z., He, X., and Zhou, D. H. (2014). A survey of event-based strategies on control and estimation. *Systems Science and Control Engineering*, 2(1):90–97.

Liu, Z. and Wang, J. (2012). Multi-scale data fusion for smart grids. In *Proceedings of the 2012 7th IEEE Conference on Industrial Electronics and Applications, ICIEA 2012*, pages 2059–2062.

Lu, X., Zhang, H., Wang, W., and Teo, K. L. (2005). Kalman filtering for multiple time-delay systems. *Automatica*, 41(8):1455–1461.

Luo, R. C. and Kay, M. G. (1989). Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):901–931.

Ma, J. and Sun, S. (2011). Optimal Linear Estimators for Systems With Random Sensor Delays , Multiple Packet Dropouts and Uncertain Observations. *IEEE Transactions on Signal Processing*, 59(11):5181–5192.

Maehlisch, M., Schweiger, R., Ritter, W., and Dietmayer, K. (2006). Multisensor vehicle tracking with the probability hypothesis density filter. In *Proceedings of the International Conference on Information Fusion*, pages 1–8.

Mahler, R. P. (2004). "Statistics 101" for multisensor, multitarget data fusion. *IEEE Aerospace and Electronic Systems Magazine*, 19(1 II):53–64.

Mamdani, E. H. and Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1–13.

Maróti, M., Kusy, B., Simon, G., and Lédeczi, Á. (2004). The Flooding Time Synchronization Protocol. In *Proceedings of International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 39–49.

Marvasti, F. (2001). *Nonuniform Sampling Theory and Practice*. Springer US, 1 edition.

Micheli, M. and Jordan, M. I. (2002). Random sampling of a continuous-time stochastic dynamical system. In *in Proc. of the 15th International Symposium on the Mathematical Theory of Networks and Systems (MTNS 2002), (University of Notre Dame, South)*.

Middleton, D. and Esposito, R. (1968). Simultaneous Optimum Detection and Estimation of Signals in Noise. *IEEE Transactions on Information Theory*, 14(3):434–444.

Mills, D. L. (1991). Internet Time Synchronization: The Network Time Protocol. *IEEE Transactions on Communications*, 39(10):1482–1493.

Mingxia Xu, Minjian Zhao, and Shiju Li (2005). Lightweight and energy efficient time synchronization for sensor network. *Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005.*, 2:947–950.

Miskowicz, M. (2006). Send-on-delta concept: An event-based data reporting strategy. *Sensors*, 6(1):49–63.

Moayedi, M., Foo, Y. K., and Soh, Y. C. (2011). Filtering for networked control systems with single/multiple measurement packets subject to multiple-step measurement delays and multiple packet dropouts. *International Journal of Systems Science*, 42(3):335–348.

Mohinder, S. G. and Angus, P. A. (2010). Applications of Kalman Filtering in Aerospace 1960 to the Present. *IEEE Control Systems Magazine*, pages 69–78.

Molchanov, I. and Molinari, F. (2014). Applications of Random Set Theory in Econometrics. *Annual Review of Economics*, 6:229–251.

Nahi, N. (1969). Optimal recursive estimation with uncertain observation. *IEEE Transactions on Information Theory*, 15(4):457–462.

Nahin, P. J. and Pokoski, J. L. (1980). NCTR Plus Sensor Fusion Equals IFFN or Can Two Plus Two Equal Five? *IEEE Transactions on Aerospace and Electronic Systems*, AES-16(3):320–337.

Nuñez-Garcia, J. and Wolkenhauer, O. (2002). Random set system identification. *IEEE Transactions on Fuzzy Systems*, 10(3):287–296.

Papoulis, A. (1984). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Inc.

Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, 1991.

Pawlak, Z. and Skowron, A. (2007). Rudiments of rough sets. *Information Sciences*, 177(1):3–27.

Peñarrocha, I., Sanchis, R., and Romero, J. A. (2012). State estimator for multisensor systems with irregular sampling and time-varying delays. *International Journal of Systems Science*, 43(8):1441–1453.

Phillips, C. L. and Nagle, H. T. (1995). *Digital control system analysis and design*. Prentice-Hall, Inc.

Ping, S. (2003). Delay Measurement Time Synchronization for Wireless Sensor Networks. *Intel Research Berkeley Lab*, pages 1–10.

Rao, N. (1998). To fuse or not to fuse: Fuser versus best classifier. *Proceedings of SPIE - The International Society for Optical Engineering*, 3376:25–34.

Richard, J.-P. (2003). Time-delay systems: an overview of some recent advances and open problems. *Automatica*, 39(10):1667–1694.

Richardson, J. M. and Marsh, K. A. (1988). Fusion of Multisensor Data. *The International Journal of Robotics Research*, 7(6):78–96.

Römer, K. (2003). Temporal Message Ordering in Wireless Sensor Networks. In *IFIP Mediterranean Workshop on Ad-Hoc Networks*, pages 131–142.

Safari, S., Shabani, F., and Simon, D. (2014). Multirate multisensor data fusion for linear systems using Kalman filters and a neural network. *Aerospace Science and Technology*, 39:465–471.

Sahebsara, M., Chen, T., and Shah, S. L. (2007). Optimal filtering with random sensor delay, multiple packet dropout and uncertain observations. *International Journal of Control*, 80(2):292–301.

Särkkä, S. (2013). *Bayesian Filtering and Smoothing*. Cambridge University Press.

Schenato, L., Sinopoli, B., Franceschetti, M., Poolla, K., and Sastry, S. S. (2007). Foundations of Control and Estimation Over Lossy Networks. *Proceedings of the IEEE*, 95(1):163–187.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.

Shen, B., Wang, Z., and Huang, T. (2016). Stabilization for sampled-data systems under noisy sampling interval. *Automatica*, 63:162–166.

Shuli Sun, Lihua Xie, Wendong Xiao, and Nan Xiao (2008). Optimal Filtering for Systems With Multiple Packet Dropouts. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 55(7):695–699.

Sichitiu, M. L. and Veerarittiphan, C. (2003). Simple, accurate time synchronization for wireless sensor networks. *IEEE Wireless Communications and Networking Conference, WCNC*, 2(C):1266–1273.

Singer, R. A. and Kanyuck, A. J. (1971). Computer control of multiple site track correlation. *Automatica*, 7(4):455–463.

Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M. I., and Sastry, S. S. (2004). Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464.

Sivrikaya, F. and Yener, B. (2004). Time Synchronization in Sensor Networks: A Survey. *IEEE Network*, 18(4):45 – 50.

Sorenson, H. W. (1970). Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum*, 7:63–68.

Srivastava, R. P. (2011). An introduction to evidential reasoning for decision making under uncertainty: Bayesian and belief function perspectives. *International Journal of Accounting Information Systems*, 12(2):126–135.

Stone, J. V. (2013). *Bayes' Rule: A Tutorial Introduction to Bayesian Analysis*. Sebtel Press.

Su, W. and Akyildiz, I. F. (2005). Time-diffusion synchronization protocols in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 13(2):384–397.

Sugeno, M. (1985). *Industrial Applications of Fuzzy Control*. Elsevier Science Inc., New York, NY, USA.

Süli, E. and Mayers, D. F. (2003). *An Introduction to Numerical Analysis*. Cambridge University Press.

Tabuada, P. (2007). Event-Triggered Real-Time Scheduling of Stabilizing Control Tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685.

Teixeira, B. O., Santillo, M. A., Erwin, R. S. T., and Bernstein, D. S. (2008). Spacecraft Tracking Using Sampled-Data kalman filters. *IEEE Control Systems*.

Teixeira, B. O. S. (2008). *Constrained state estimation for linear and nonlinear dynamic systems*. PhD thesis, Universidade Federal de Minas Gerais.

Theil, A., Kester, L. J., and Bossé, É. (2000). On measures of performance to assess sensor fusion effectiveness. *Proceedings of the 3rd International Conference on Information Fusion, FUSION 2000*, 2:3–7.

van Greunen, J. and Rabaey, J. (2003). Lightweight Time Synchronization for Sensor Networks. In *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications(WSNA)*, pages 11–19.

Wahlström, N., Axelsson, P., and Gustafsson, F. (2014). Discretizing stochastic dynamical systems using Lyapunov equations. *arXiv e-prints*, page arXiv:1402.1358.

Wang, Z., Ho, D. W., Liu, Y., and Liu, X. (2009). Robust H_∞ control for a class of nonlinear discrete time-delay stochastic systems with missing measurements. *Automatica*, 45(3):684–691.

Westenberger, A., Waldele, S., Dora, B., Duraisamy, B., Muntzinger, M., and Dietmayer, K. (2013). Multi-sensor fusion with out-of-sequence measurements for vehicle environment perception. In *Proceedings - IEEE International Conference on Robotics and Automation*.

Wiener, N. (1949). *The Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Wiley.

Willner, D., Chang, C. B., and Dunn, K. P. (1976). Kalman filter algorithms for a multi-sensor system. In *IEEE Conference on Decision and Control including the 15th Symposium on Adaptive Processes*, volume 15, pages 570–574.

Yan, B., Lev-Ari, H., and Stankovic, A. M. (2017). Networked State Estimation with Delayed and Irregularly-Spaced Time-Stamped Observations. *IEEE Transactions on Control of Network Systems*, 5870(c):1–1.

Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control*, 8:338–353.

Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3–28.

Zhu, C., Xia, Y., Xie, L., and Yan, L. (2013). Optimal linear estimation for systems with transmission delays and packet dropouts. *IET Signal Processing*, 7(9):814–823.

Zou, L., Wang, Z.-D., and Zhou, D.-H. (2017). Event-based Control and Filtering of Networked Systems: A Survey. *International Journal of Automation and Computing*, 14(3):239–253.