

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

Modeling, Analysis and Control of Nonlinear Systems Laboratory
Electronics Engineering Department
Federal University of Minas Gerais
Av. Antônio Carlos 6627, 31270-901 Belo Horizonte, MG Brasil
Phone: +55 3499-4866 - Fax: +55 3499-4850



Sensor Fusion for Irregularly Sampled Systems

Taiguara Melo Tupinambás

Master dissertation submitted to the Graduate Program in Electrical Engineering of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

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

Belo Horizonte, February, 2019

Agradecimentos

Antes de tudo, sou grato à UFMG por ter me proporcionado toda uma estrutura de excelência, para a realização dos meus estudos. É através de instituições públicas de ensino superior como ela, que podemos evoluir como sociedade. Estendo meus agradecimentos ao PPGEE-UFMG e aos seus funcionários, professores e alunos que contribuíram para o desenvolvimento deste trabalho. E ao CNPq, pelo apoio financeiro durante grande parte do meu mestrado.

Agradeço aos meus orientadores, Prof. Bruno Teixeira e Prof. Leonardo Tôrres, por todo o ensinamento e pela forma contagiente com que compartilham suas experiências. Serei eternamente grato por terem me acolhido de volta à Academia, após longos anos.

Aos amigos do Laboratório de Modelagem, Análise e Controle de Sistemas Não-Lineares (MACSIN), pelo companheirismo e apoio. Aprender junto com vocês foi incrível.

Também tenho muito a agradecer à equipe do Laboratório de Nanoespectroscopia (LabNS), à qual me juntei recentemente, e que tem renovado de forma constante a minha paixão pela ciência.

Às amigas e aos amigos do Bloco Pré-Sal, por trazerem leveza e alegria à minha vida, além de serem modelos exemplares que tenho para a vida. À turma de graduação da Elétrica, em especial ao Igor Baratta, grande amigo e conselheiro.

Termino esses agradecimentos, mencionando aqueles sem os quais nada faria sentido nem seria possível. Aos meus pais, Marilu e Ubiratan, e à minha irmã, Moara, agradeço imensamente pela formação, pela confiança, pelos conselhos e pelo amor incondicional. Vocês me fazem muito feliz. À minha amada esposa Marina, meu porto seguro. Você me faz ser uma pessoa melhor todo dia. E às minhas famílias Melo e Tupinambás, pelo amor e pelo incentivo constantes.

Epigraph

*"Vamos colocar nas mãos do índio
os botões da informática.*

*Vamos preparar com raio laser
uma grande feijoada."*

—Carlos Fernando, *Pátria Amada*

Contents

Resumo	xi
Abstract	xiii
List of Figures	xv
List of Tables	xvii
List of Symbols and Acronyms	xix
1 Introduction	1
1.1 Motivation	1
1.2 Problem Formulation	4
1.3 Objectives	5
1.4 Text Outline	6
2 Sensor Fusion	9
2.1 Contextualization	9
2.2 Motivation and Advantages	10
2.3 Taxonomy and Classification	11
2.4 Approaches for the Fusion of Imperfect Data	17
2.4.1 Probabilistic	18
2.4.2 Evidential Belief	19
2.4.3 Fuzzy Logic	20
2.4.4 Possibilistic	21

2.4.5 Random Set	22
2.4.6 Rough Set	24
2.5 Chapter Summary and Final Remarks	25
3 Irregular Sampling	27
3.1 Definition of Irregular Sampling	27
3.2 Contextualization	28
3.3 Types of Sampling Irregularity	30
3.3.1 Time Delay	30
3.3.2 Packet Loss	32
3.3.3 Uncertain Observation	33
3.3.4 Aperiodic Sampling	34
3.3.5 Multi-Rate Sampling	38
3.4 Time Synchronization	40
3.5 Chapter Summary and Final Remarks	43
4 State Estimation for Sampled-Data Systems	45
4.1 Discrete-Time Model of Sampled-Data Systems	45
4.2 Bayesian Estimation	49
4.2.1 Kalman Filter	54
4.2.2 Unscented Kalman Filter	59
4.3 State Estimation with Aperiodic Sampling	61
4.3.1 Aperiodic Sampling as a Poisson Process	63
4.3.2 State Estimation With Timestamp	65
4.3.3 State Estimation Without Timestamp	67
4.4 Performance Metrics	69
5 Numerical Results	73
5.1 Error Effects Introduced by Shifting Time Instants	73
5.2 Design of Simulation Setup	76
5.3 Linear System	77
5.3.1 System Description	78

5.3.2	Single Realization Analysis	82
5.3.3	Measurement Signal-to-Noise Ratio Variation	85
5.3.4	Average Sampling Rate Variation	86
5.3.5	Regular and Average Irregular Time Interval Relation Variation .	87
5.4	Nonlinear System: Unicycle Position	90
5.4.1	System Description	90
5.4.2	Single Realization Analysis	91
5.4.3	Measurement Signal-to-Noise Ratio Variation	94
5.4.4	Average Sampling Rate Variation	94
5.4.5	Regular and Average Irregular Time Interval Relation Variation .	95
5.5	Simulated Results Discussion	95
6	Conclusions	99
6.1	Project Overview	99
6.2	Main Results and Contributions	100
6.3	Future Work	101
References		112

x

Resumo

O uso de vários sensores para melhorar a qualidade na informação obtida pelos dados tem crescido de forma contínua nas últimas décadas. Com os avanços em tecnologia de microprocessadores e dispositivos de comunicação, redes de sensores continuarão a crescer em tamanho e complexidade. As aplicações mais populares para combinar dados de múltiplos sensores estão relacionadas a estimativa de estados de um sistema dinâmico. Para isso, duas fontes ruidosas de informação são necessárias: um modelo de processo, que descreve como os estados evoluem no tempo; e um modelo de observação, cujos dados geralmente provém de sensores. Como a maioria dos sensores são digitais, os sinais devem ser amostrados para que possam ser processados, dando origem aos sistemas amostrados. Estimadores de estados clássicos para esses casos, como o famoso filtro de Kalman, consideram implicitamente amostragem regular de sinais, com intervalo de tempo constante entre amostras, de forma que sistemas em tempo contínuo podem ser discretizados em representações invariantes no tempo, na maioria dos casos. No entanto, devido ao cada vez mais comum uso de complexas redes de sensor sem sincronização temporal explícita, muitas aplicações não podem depender de dados transmitidos de forma regular. Existem adaptações aos métodos de estimativa de estados para lidar com a maioria das irregularidades, desde que os carimbos de tempo sejam parte do pacote de medição e que o aumento no custo computacional seja aceitável. Caso o carimbo de tempo não possa ser utilizado no processo de estimativa, pode-se investir em sincronização dos dados ou aceitar que a assimilação das informações seja feita em instantes de tempo incorretos. Os efeitos no desempenho da estimativa da última abordagem ainda é pouco estudada. Nesse trabalho, investigamos como o desempenho é deteriorado com o negligenciamento dos carimbos de tempo das medições em algoritmos de estimativa de estados. Nós consideramos o processo de Poisson como modelo para gerar a sequência de instantes de tempo irregular, e estudamos os resultados da estimativa de estados para um sistema linear e outro não-linear, simulados com amostragem aperiódica, usa utilizando o filtro de Kalman para o caso linear e sua variação *unscented* para o caso não-linear. Algoritmos são implementados tanto para utilizar quanto para negligenciar o carimbo de tempo no processo de estimativa e os resultados de várias realizações são comparados para diferentes cenários de simulação. Finalmente, identificamos e discutimos a relação entre diferentes conjuntos de parâmetros, como níveis de sinal-ruído e frequências médias de amostragem, e os efeitos no desempenho da estimativa.

Keywords: Fusão Sensorial; Amostragem Irregular; Estimação de Estados; Sistemas Amostrados; Carimbo de Tempo.

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 obtained from sensors. Since most sensors are digital, signals must be sampled in order to be processed, leading to sampled-data systems. Classical state estimators in these cases, like the well-known Kalman filter, implicitly consider regularly sampled signals with constant time intervals between samples, such that continuous-time systems can be time discretized into time-invariant representations in most cases. However, because of the widespread use of complex sensor networks without explicit time synchronization, many applications cannot rely on data being transmitted regularly. There are adaptations to state estimation techniques that handle most of the irregularities, provided that timestamps are part of measurement packets and that the increase in computational processing time is acceptable. If timestamps cannot be used in the estimation process, one can either invest in synchronization or accept the assimilation of information at incorrect time instants. The effects in estimation performance of the latter approach has not yet been extensively studied. In this work we investigate how performance is deteriorated by neglecting measurements timestamps in state estimation algorithms. We consider the Poisson process as a model to generate the irregular time instants sequences, and we assess state estimation results for linear and nonlinear systems simulated with aperiodic sampling, using the Kalman filter for the former and its adapted unscented version for the latter. Algorithms are designed to use timestamps or to neglect them in the estimation process, and their results over multiple runs are compared for different simulation scenarios. Finally, we identify and discuss relations between different sets of parameters, such as signal-to-noise ratios and average sampling frequencies, and the degradation in performance.

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

List of Figures

1.1	Irregular sampling process modeled by a Poisson random process	5
2.1	PDFs for different amount of fused measurements	11
2.2	Expected advantages in sensor fusion	12
2.3	Classification of data fusion based on sensor interaction.	13
2.4	Input and output model and the three fusion levels.	14
2.5	Sensor network architectures for data fusion.	15
2.6	Data fusion categorization based on data challenges hierarchy	17
2.7	Fuzzy inference system example	21
2.8	Rough set approximation by lower and upper crisp sets	25
3.1	Irregular sampling diagram	29
3.2	Time delay diagram	31
3.3	Different classes of out-of-sequence measurement irregularities	31
3.4	Packet loss diagram	32
3.5	Uncertain observations diagram	34
3.6	Aperiodic sampling diagram	35
3.7	Time-driven and event-driven sampling schemes	36
3.8	Two models of aperiodic sampling	37
3.9	Multi-rate sampling diagram	39
3.10	Periodic and aperiodic multi-rate sampling scheme	40
4.1	<i>Posterior PDF obtained by the fusion prior and likelihood</i>	55
4.2	Input and observation time instants realization	62
4.3	Arrivals of a Poisson Process	63
4.4	Illustrative schematic of the <i>online</i> estimator, with timestamp	66
4.5	Linear simulation results for G(s)	68
4.6	Periodic and aperiodic sampling of an LTI output	69

4.7	Effect of approximating the random time instants of the aperiodic sampling.	69
4.8	χ^2 PDFs and acceptance regions	71
5.1	Added error of shifting measurements time instants	74
5.2	Added error of shifting measurements time instants	75
5.3	QQ-plot of the measurement error	75
5.4	Surface plot of estimated RMSE from time shifting error, as a function of SNR and λ	76
5.5	Bode diagram of both modes	79
5.6	PRBS samples used as input	80
5.7	Linear simulation results for PRBS input in different time frames	81
5.8	Output of the fourth order system to the PRBS input signal	82
5.9	Estimated and true states	83
5.10	Performance index temporal cut for one realization	84
5.11	Consistency tests for linear system	85
5.12	Linear system estimation performance comparison	89
5.13	Nonholonomic robot system representation	91
5.14	Simulation input signals	92
5.15	Single realization of unicycle position estimation	93
5.16	Nonlinear system estimation performance comparison	97

List of Tables

2.1	Possibility and Probability functions associated with X	22
2.2	Mathematical parallels between single and multi sensor statistics	23
2.3	Data fusion methods for imperfect data	26
3.1	Consistency test results for fusion of time delayed measurements with uncertain timestamps	41
3.2	Comparison of time synchronization methods	44
5.1	Review of concepts and symbols used in simulation	78
5.2	Sets of parameters for linear system simulation	84
5.3	Linear system paired t-test results and effect size estimates for RMSE of states x_1 and x_4	88
5.4	Sets of parameters for nonlinear system simulation parameters	93
5.5	Nonlinear system paired t-test results and effect size estimates for position RMSE	96

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	i th 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;
$v(t_k)$	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 i^{th} rule;
y_i	i^{th} consequent output variable for the i^{th} rule;
C_i	consequent class for the i^{th} 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 ;
\subseteq	is a subset of;
\cap	set intersection;
\emptyset	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;
\forall	for all;
$>$	greater than;
\geq	greater than or equal to;
\sim	is distributed as;
\pm	plus or minus;
y_t, y_k, y_{kT}	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;
l	amount of different known delays;
$\xi(t)$	Bernoulli random variable
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{N}	natural numbers;
\mathbb{R}	real numbers;
\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;
\forall	for all;
$<$	less than;
\leq	less than or equal to;
$\arg \max_x f(x)$	argument x that maximizes function $f(x)$;
\xrightarrow{D}	converges in distribution;
$F_X(x)$	cumulative distribution of the random variable X ;
$\bar{F}_X(x)$	complementary cumulative distribution of the random variable X ;
$\rho(x)$	probability density function of x ;
$\rho(x y)$	conditional probability density function of X given Y ;
$\hat{\rho}(x y)$	estimated conditional probability density function of X given Y ;
$\arg \max_{x_k}$	the argument of the domain x_k of the maxima
$\arg \min_{x_k}$	the argument of the domain x_k of the minimum
I	identity matrix;
A^T	transpose of A ;
A^{-1}	inverse of A ;
f	process model;
f_d	discrete-time process model;
f_d^*	time-invariant discrete-time process model;
g	observation model;
t	continuous-time index;
k	discrete-time index of measurements;
j	discrete-time index of estimation;
i	discrete-time index of input;
t_k	continuous-time sampled instants;
t_j^*	state estimation time sequence;
δt_j^*	variable time interval;
T	input sampling interval;
A, B, C, G	matrices of continuous-time linear model in state-space;
A_d, B_d, C_d, G_d	matrices of discrete-time linear model in state-space;
A_k, B_k, C_k, G_k	time-varying matrices of discrete-time linear model in state-space;
$x(t)$	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;
$u(t_k)$	input vector at continuous-time sampled instants;
$u(iT)$	input vector at continuous-time sampled at time $t = iT$;
$w(t)$	process noise vector;
$w(t_k)$	process noise vector at continuous-time sampled instants;
$w_d(t_k)$	process noise vector at continuous-time sampled instants;
$y(t_k)$	output vector at continuous-time sampled instants;

$v(t_k)$	measurement noise vector;
z_k	constant evidence term;
Q	process noise covariance matrix;
R	measurement noise covariance matrix;
h_k	time intervals between two continuous-time sampled instants;
k_i	i -th slope approximation for Runge-Kutta method;
τ	integration auxiliary variable;
σ	standard deviation;
$\hat{\sigma}_p$	estimated standard deviation of the prior;
$\hat{\sigma}_e$	estimated standard deviation of the evidence;
\hat{x}_p	estimated mean of the prior;
μ_{fused}	estimated mean of the fused information;
σ_{fused}	estimated standard deviation of the fused information;
Σ	covariance matrix;
P^{xx}	covariance matrix of x ;
P^{yy}	covariance matrix of y ;
P^{xy}	cross-covariance matrix of x and y ;
\bar{x}	mean of x ;
$x_{k k}$	data-assimilation state estimate;
$x_{k k-1}$	forecast state estimate;
$P_{k k-1}^{xx}$	forecast error covariance matrix;
$P_{k k}^{xx}$	data-assimilation covariance matrix;
$P_{k k}^{yy}$	innovation covariance matrix;
$P_{k k}^{xy}$	cross-covariance matrix;
K_k	Kalman gain;
χ_i	i^{th} column of the sigma-point matrix;
γ_i	i^{th} weight related to the sigma-point matrix;
Υ_i	propagated sigma-point column;
ψ_{UT}	unscented transform function;
d	deterministic vector parameter of unscented transform;
λ	average sampling rate of observations;
α	relation between T and $1/\lambda$;
$N(t)$	amount of measurements arrived in the interval t ;
ξ_i	phase of each sensor in a network;
$G(s)$	LTI system transfer function;
s	Laplace variable;
η	innovation vector;
μ_D	mean difference of the t-test;
d	cohen's d effect size for the mean difference;
s_D	standard deviation of the paired-samples difference.

Chapter 5

\mathbb{N}	natural numbers;
\mathbb{R}	real numbers;
\mathbb{R}^n	\mathbb{R}^{nx1} n -dimensional Euclidean space;
\forall	for all;
\in	belongs to;
\sim	is distributed as;
\triangleq	equals by definition;
\approx	approximately equal to;
$I_{m \times n}$	m by n identity matrix;
t	continuous-time index;
k, i	discrete-time index;
n	discrete-time index of time-shifted measurements;
T	input sampling interval;
t_k, t_i	continuous-time sampled instants;
δ_k	shifted time interval;
δt_{sim}	nominal system simulation time step;
h_k	time intervals between two continuous-time sampled instants;
$A, B, C, D,$	matrices of continuous-time linear model in state-space;
A_d, B_d, D_d, D_d	matrices of discrete-time linear model in state-space;
$G(s)$	LTI system transfer function;
$G_{lp}(s), G_{hp}(s)$	low-pass and high-pass LTI system transfer function, respectively;
s	Laplace variable;
τ_{lp}, τ_{hp}	low-pass and high-pass system time constant, respectively;
ω_{lp}, ω_{hp}	low-pass and high-pass system natural frequency, respectively;
ζ_{lp}, ζ_{hp}	low-pass and high-pass system damping frequency, respectively;
$x(t)$	state vector;
x_i	i^{th} state of the state vector;
$x(t_k)$	state-vector at continuous-time sampled instants;
$\dot{x}(t)$	derivative of the state vector with respect to t ;
$u(t)$	input vector;
\tilde{u}	measured input vector;
$y(t)$	output vector;
$y(t_k)$	output vector at continuous-time sampled instants;
$w(t_k)$	process noise vector at continuous-time sampled instants;
$v(t_k)$	measurement noise vector at continuous-time sampled instants;
Q	process noise covariance matrix;
R	measurement noise covariance matrix;
e	error introduced by shifting sample time instants;
η	innovation vector;
μ_D	mean difference of the t-test;

d	cohen's d effect size for the mean difference;
λ	average sampling rate of observations;
α	relation T and $1/\lambda$;
p_x, p_y	position coordinates in the xy -plane;
\dot{p}_x, \dot{p}_y	derivative of the position coordinates;
θ	angular orientation;
$\dot{\theta}$	derivative of angular orientation;
v	linear velocity;
\dot{v}	derivative of linear velocity, or linear acceleration;
N	total amount of estimates.

Acronyms

CDF	Cumulative Distribution Function;
CI	Confidence Interval;
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;
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;
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	Pseudo-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;
UT	Unscented Transform;
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, proba-

bilistic 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 instrumen-

tation 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 ad-

ditional 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 timestamp for different scenarios. The purpose is to shed some light on the trade-off for investments in sensor networks and synchronization, proposing a framework to assist the decision making process.

1.2 Problem Formulation

Consider the stochastic nonlinear sampled system

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

$$y(t_k) = g(x(t_k), v(t_k), t_k), \quad (1.2)$$

where $f: \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}^+ \rightarrow \mathbb{R}^n$ and $g: \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^+ \rightarrow \mathbb{R}^m$ 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 a strictly increasing sequence ($t_{k+1} > t_k, \forall k \in \mathbb{N}^+$) and defined by the time intervals $h_k \triangleq t_k - t_{k-1}, \forall k \geq 1$. 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) considered a set of N identical sensors measuring the state variables of a physical process every L seconds. They have shown 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 $\lambda = N/L$.

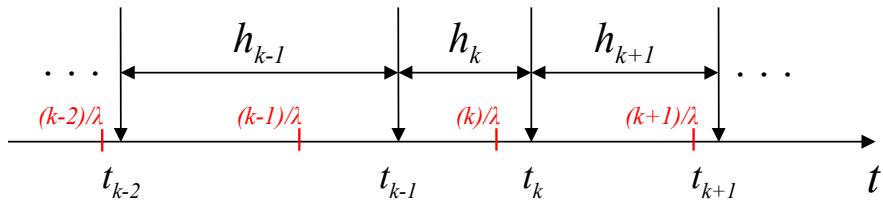


Figure 1.1: Irregular sampling process modeled by a Poisson random process. Regularly spaced time intervals $1/\lambda$ are shown in red. An example of time instants t_k realization is also shown, with the respective random time intervals h_k . The expected value of time interval is given by $E[h_k] = \frac{1}{\lambda}$.

When time-stamp information is available, data assimilation can be performed considering the correct measurement instants t_k . When they are not, the assimilation moment is assumed to be the random reception 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

In this study our main goal is to assess the importance of the timestamp in sensor fusion for the problem of irregular sampling. More specifically, we define the following research objectives:

1. Review sensor fusion methods and the irregular sampling problem;
2. Discuss state estimation algorithms that fuse data from multiple sources, subject to irregularly sampled measurements with no information about timestamp;
3. Develop a simulation setup where the effect of neglecting timestamps in state estimation can be assessed;
4. Apply the proposed setup to linear and nonlinear systems, considering performance metrics that assess accuracy, which is related to estimation quality; and consistency, which quantifies conformity of the estimation error and the covariance matrix provided by the filter.

1.4 Text Outline

This text is organized in six chapters, including this one, which presents an overview of the motivations and historical perspective of the theme, objectives and problem formulation.

Chapter 2 presents a review of sensor fusion, 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, which is divided in four groups: imperfection, correlation, inconsistency and disparateness. We then introduce and discuss the methods proposed in literature to handle imperfect data, the category that encompasses information uncertainty, imprecision and granularity.

Chapter 3 discusses the data-related challenges that arise for sampled-data systems, regarding sampling irregularities. First we define the irregular sampling problem. Then 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 exact time instants when 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.

In Chapter 4 we describe the discrete-time representation of the sampled-data systems. Then we present the methods used in the simulations, considering the adaptations for the scenarios with and without timestamp. We develop the filtering algorithm and the assumptions used for each scenario. Furthermore, we define the performance metrics used for the results assessment.

We start Chapter 5 by studying the error effect introduced in the measurements by neglecting timestamp and shifting time instants. Then we discuss the simulation setup and present the 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, with and without time-stamp information in estimation algorithms. Performance is evaluated using estimated state errors and estimation consistency.

Finally, we conclude the study in Chapter 6, providing an overview of the work, 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 in 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 classification models for fusion methods. The categorization based on data challenges, that is imperfection, correlation, 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 probably 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. However, such improvement depends on the condition that the statistical properties and characteristics of the uncertainty sources are properly known by the fusion process.

A direct answer to Fowler's work came in the year after with the publication of the work (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, 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.

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 uncertainty levels - are shown for various quantities of random variables being averaged.

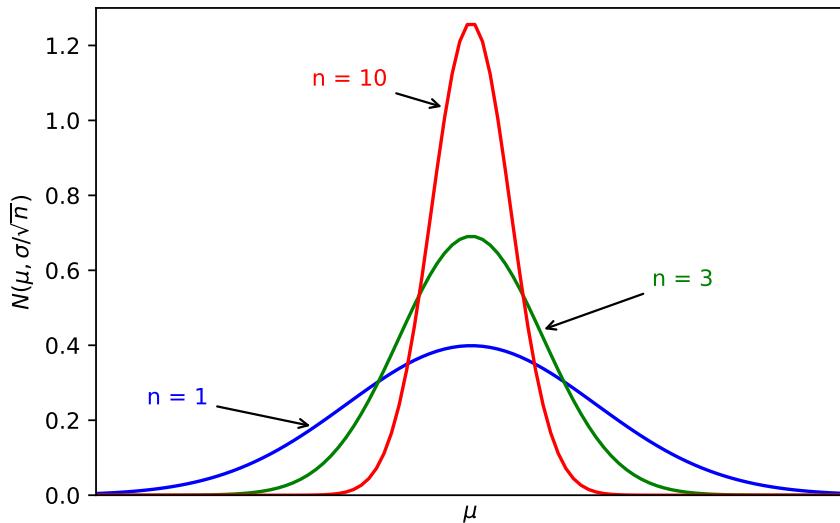


Figure 2.1: PDFs for different amount of fused measurements, obtained by 1 (blue), 3 (green) and 10 (red) redundant sensors, considering i.i.d additive Gaussian noise. The higher the value of n , the lower is the standard deviation.

There are many other advantages to fusing information from multiple sensors. Based on the studies by Hall, Elmenreich and Andler, we can group most of them 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.

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 comprehen-

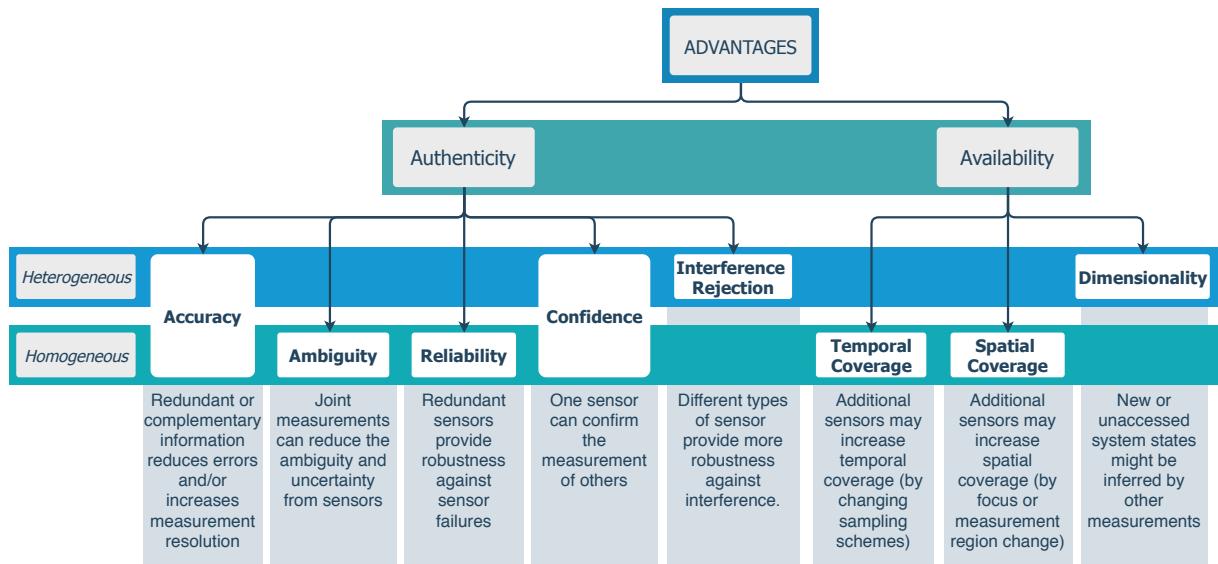


Figure 2.2: Sensor fusion categorization hierarchy based on expected advantages. The white boxes represent the advantages grouped two ways: between authenticity or availability improvements; and between heterogeneous, homogeneous or both sensor architectures

hensive 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 il-

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.

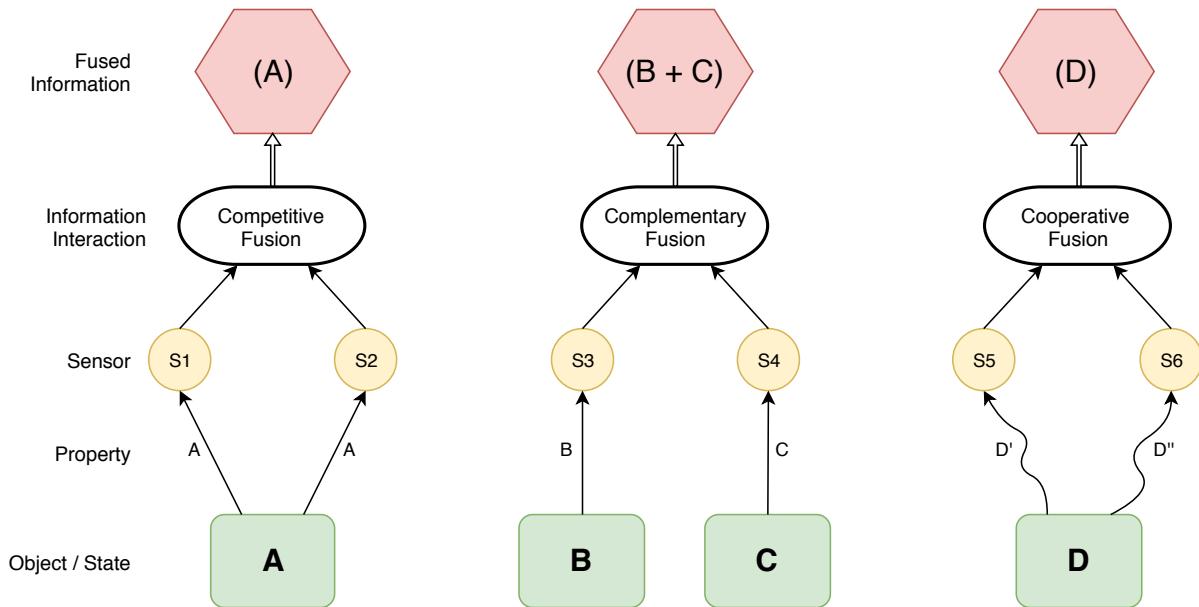


Figure 2.3: Classification of data fusion based on sensor interaction. The lower green boxes represent the states or objects being measured. Their properties can be observed directly (straight lines) or indirectly (curved lines) by sensors indicated by the letter S. After fusion, the output information is presented in the red hexagons. Adapted from ([Elmenreich, 2002](#)).

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 - decision*

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.

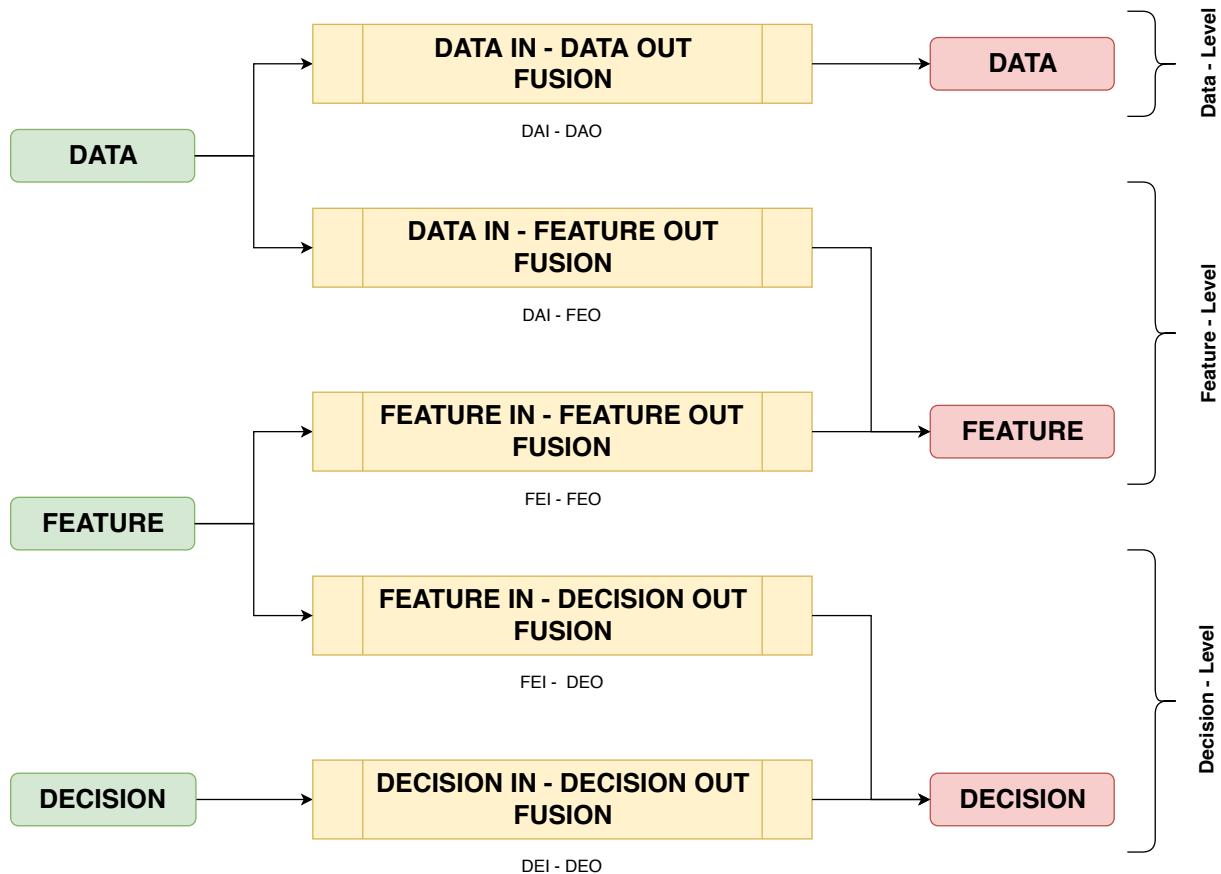


Figure 2.4: Input and output model categorization, representing type of input in green, the method in yellow and the output in red. The fusion levels are shown in the curly brackets. Adapted from ([Dasarathy, 1997](#))

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 associa-

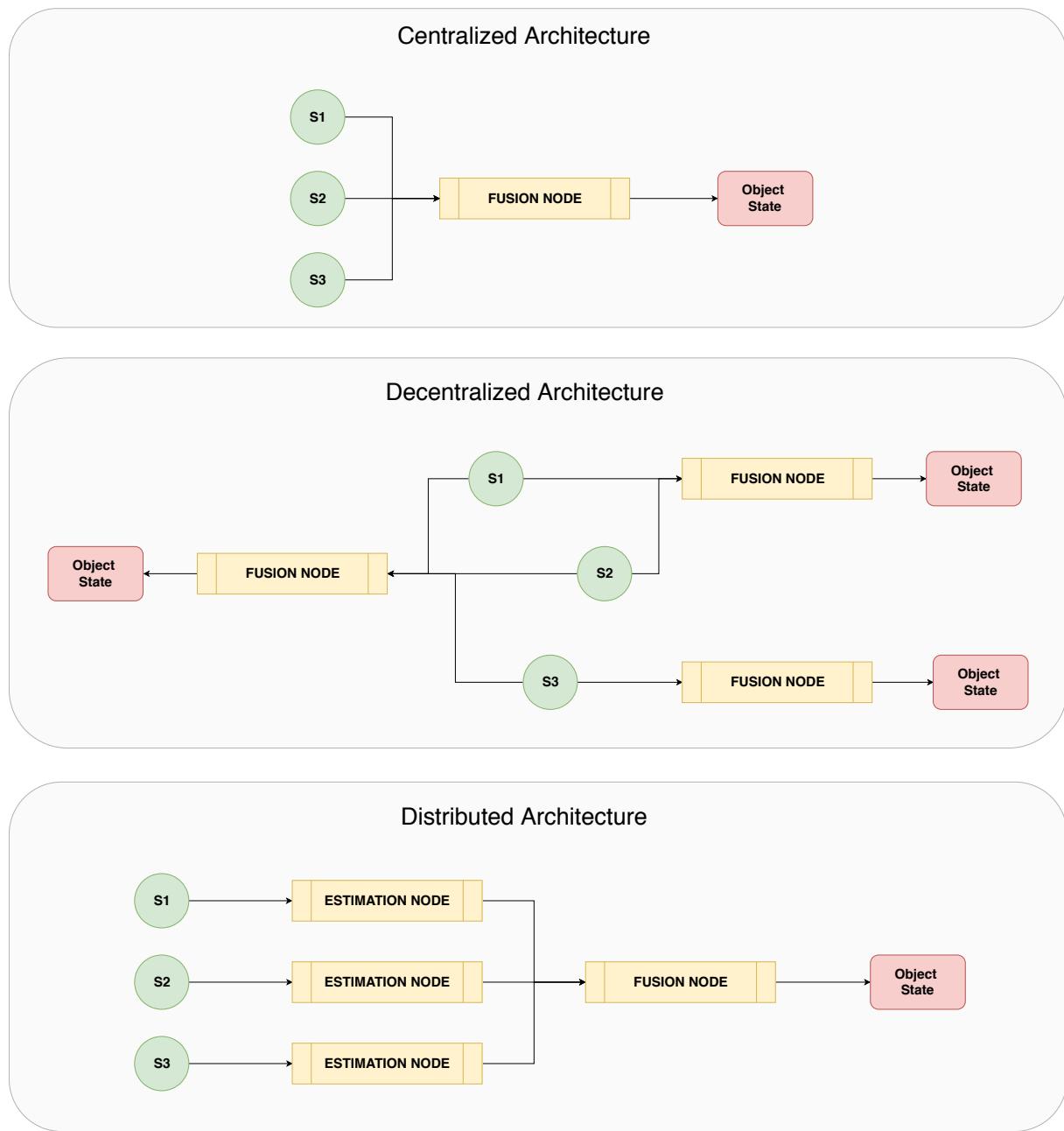


Figure 2.5: Sensor network architectures for data fusion: *centralized*, *decentralized* and *distributed*. Sensors are shown in green circles and the arrows represent the information flow. Adapted from ([Castanedo, 2013](#))

tion, such scheme should be optimal. However it is keen to many sampling irregularity related 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 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.

A final note on the taxonomy of data fusion methodologies will be given considering the work of (Khaleghi et al., 2013), due to its connections to sensor fusion in the presence of irregularities, like the ones that we discuss 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* and *disparate*. The most fundamental problem present in data is imperfection, an issue explained in details in Section 2.4. 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.

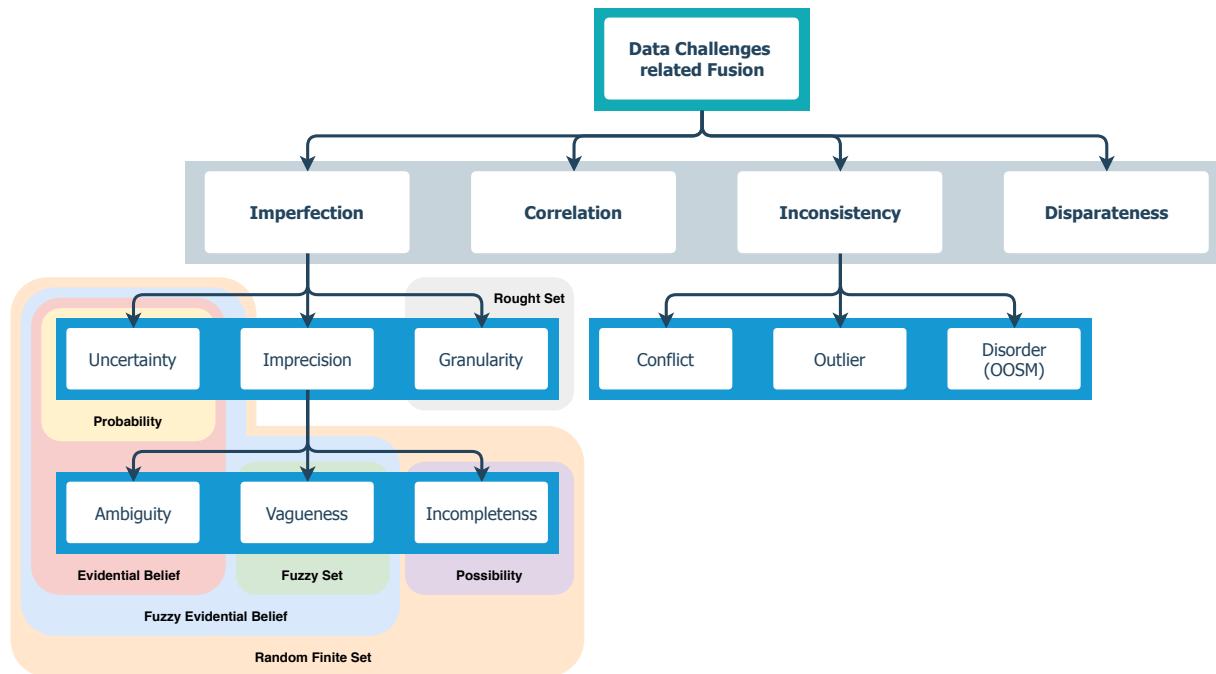


Figure 2.6: Categorization based on data challenges hierarchy. For imperfect data, fusion methods and the issues they deal with are also presented. Based on ([Khaleghi et al., 2013](#))

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. A source 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 imperfection 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 the famous equation (Stone, 2013)

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

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 the normalized likelihood of the evidence ($\rho(E|H)/\rho(E)$) to obtain 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 1942¹ 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

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

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 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 a well known to handle vagueness in information. Unlike classical crisp definitions, where an element x belongs or not to some set A , fuzzy sets are characterized by a *membership function* $\mu_A(x)$ which associate values between 0 and 1 with the degree to which a given object x belongs to a set A . That is, the closer $\mu_A(x)$ is to the one, more certain we are that x belongs to A . For example, 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 then be 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. One 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 functions of the input variables. Mamdani and Sugeno-type rules are given, respectively, by

$$R_i^{Mamdan} : \text{If } x_1 \text{ is } A_{i,1} \text{ and } x_2 \text{ is } A_{i,2} \dots \text{ and } x_n \text{ is } A_{i,n}, \text{ Then } y_i \text{ is } C_i, \quad (2.2)$$

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

where $x_j \in \mathbb{R} \forall j = 1, \dots, n$, is the j^{th} input variable, $A_{i,j}$ is the j^{th} antecedent fuzzy set for the i^{th} rule, C_i is the consequent fuzzy set for the i^{th} rule, $f_i(X)$ is the function of 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 implications as the consequents. Finally we aggregate the consequents across 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 (Yen, 1990).

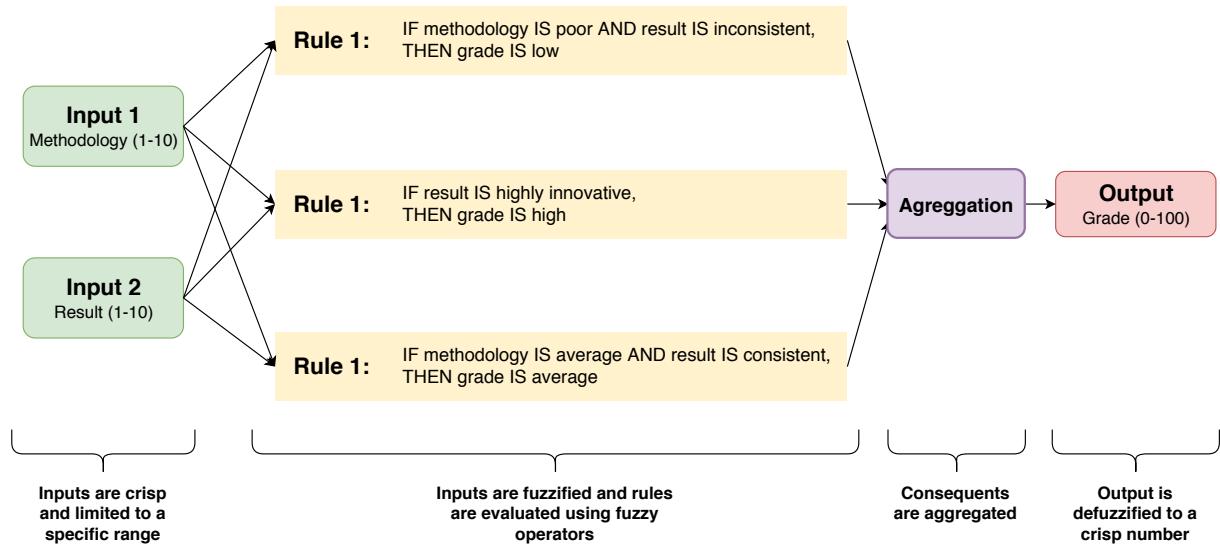


Figure 2.7: Fuzzy inference system example for grading thesis. There are two inputs: methodology and result. They are fuzzified according to the antecedent membership fuzzy sets: poor and average for methodology; and inconsistent, consistent and highly innovative for result. After evaluation of the rules, 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.

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 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 $U = 1,2,3,\dots$. 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

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

Note that according to Table 2.1 it is perfectly possible for Maria to have failed 1, 2, 3 or 4 times. The degree of possibility decreases for higher values of failure times. On the other hand, based on recent historical data on students failures, the most likely number of times for Maria to have failed is 2, whereas since no student failed more than 6 times, the probability that Maria will fail more than 6 times 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

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,
prior PDF	global PDF,
likelihood	global likelihood,
information theory	multi-target information theory
filtering theory	multi-target filtering theory

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

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

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 rough 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 B , 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 , $B(x)$ denotes the B -elementary set for an element $x \in X$, $B_*(X)$ 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 by upper and lower bounds.

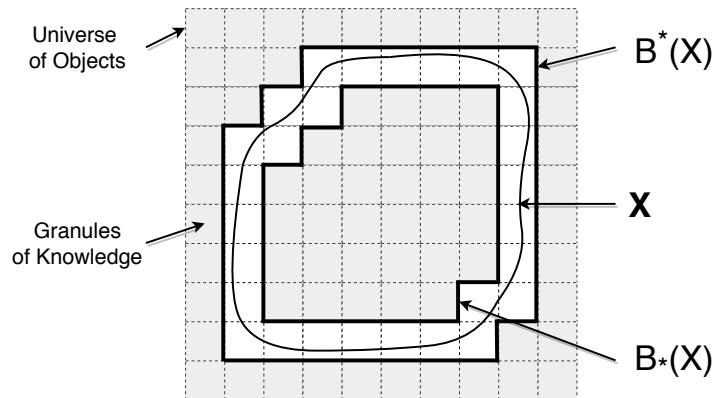


Figure 2.8: Best approximation of the rough set X by lower $B_*(X)$ and upper $B^*(X)$ crisp sets. The elementary granules are squares and the union of these granules form the universe of objects.

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.

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.

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

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

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 obtained by fusing data from multiple sensors, some challenges will naturally appear. For the state estimation problem for sampled-data systems, a common challenge is related to sampling irregularities introduced in a 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 sampling irregularities and discuss their main causes and characteristics. Then, in Section 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 a 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 continuous-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 outputs a constant value equals to the signal value at the sampling instant during the whole time interval, until the next sample is available.

Thus the state observations of a sampled-data system can be modeled as

$$y(t) = g(x(kT), v(kT), kT), \quad \text{for } kT \leq t < (k+1)T, \quad (3.1)$$

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

Therefore, a sampled-data system is regularly sampled if its observation model can be modeled by (3.1), 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.

The use of imperfect communication networks seems 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 medium (Sahelsara 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.

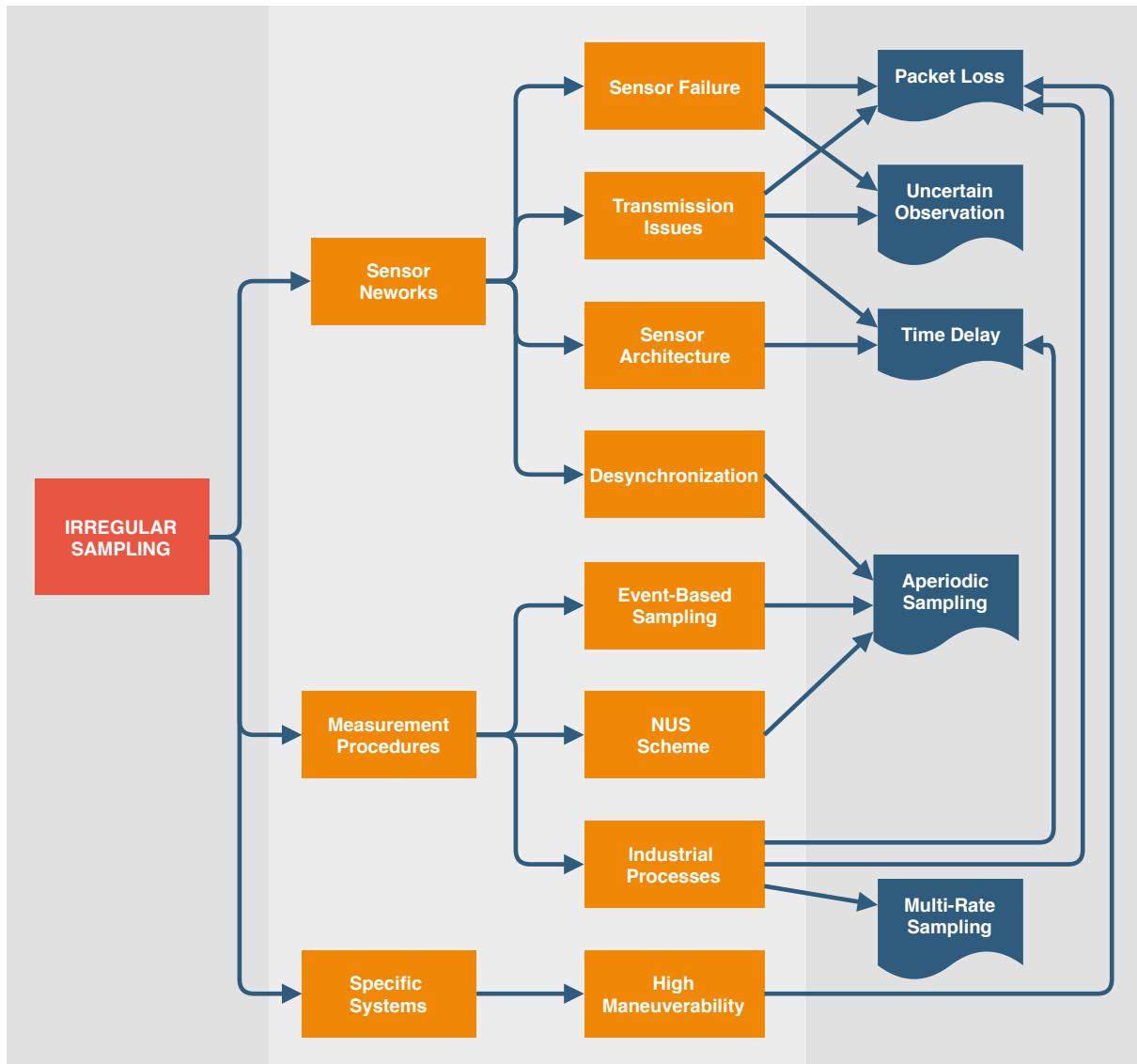


Figure 3.1: Irregular sampling diagram, showing the main causes (in orange) and effects (in blue)

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, laboratory analyses are used to improve estimation quality, but usually at slower rates, sometimes irregularly and with possible time delays (Fatehi 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 properly associated to the same observed phenomenon in order to be fused into knowledge. A crucial part of this association is the temporal synchronization of information on the time when the observations were taken (Ping, 2003). Most sensor fusion methods for irregularly sampled systems rely on the correct timestamps to develop modifications in classical algorithms. If observations are imprecisely time stamped, some alternatives have been proposed to incorporate some aspects of that imprecision in the estimation method (Julier and Uhlmann, 2005; Huck et al., 2011). Still, some knowledge about the irregularity is assumed to be known. Alternatively, many techniques for time synchronization can be performed, to ensure a common temporal reference for all the sensors.

In 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

Signal time delays might affect state estimation in various ways and due to different reasons, as shown in Figure 3.2. In some cases, delays can be modeled as constant and known (Lu et al., 2005). Alternatively they 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. State estimation algorithms can be adapted to handle such irregularity by using only the latest measurement and discarding all others, or implementing 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).

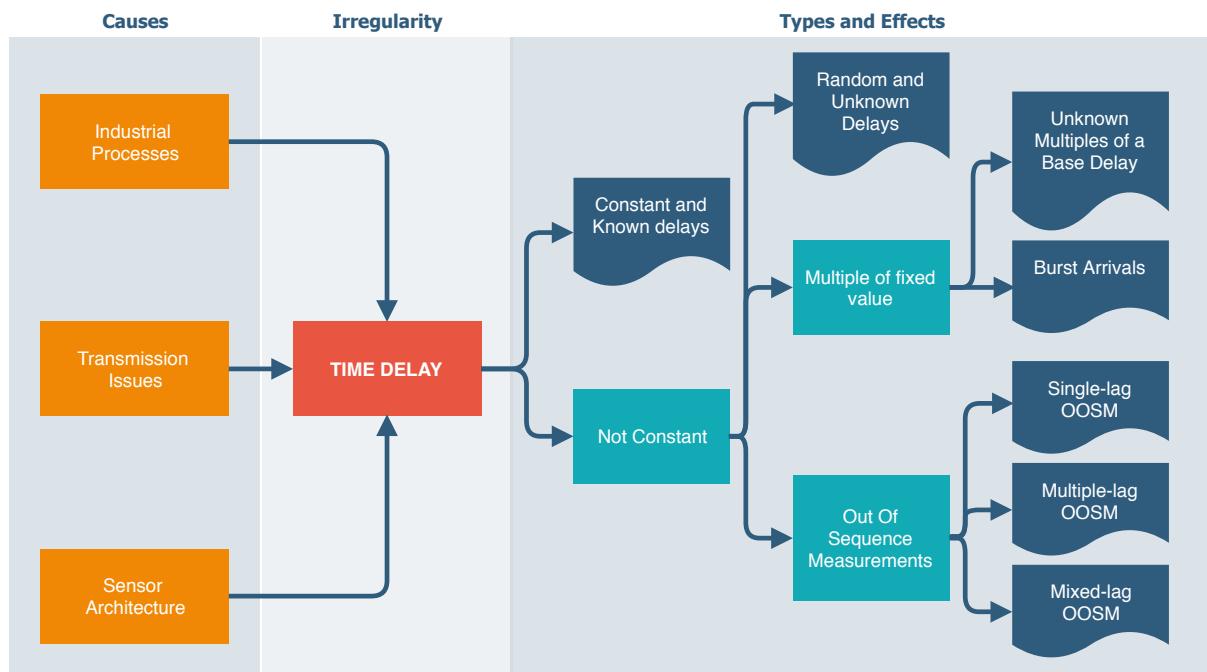


Figure 3.2: Time delay diagram, showing the main causes (in orange) and effects (in dark blue) of time delay irregularity. The light blue boxes indicate different types that lead to different effects.

Multiples of a known lag or not, delayed measurements from a multisensor system can arrive disordedly, 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.

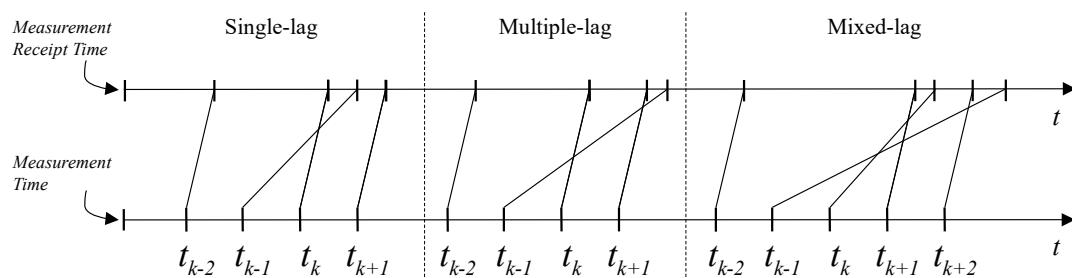


Figure 3.3: Different classes of out-of-sequence measurement irregularities

(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 scarce 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 associated with packet loss are shown in Figure 3.4.

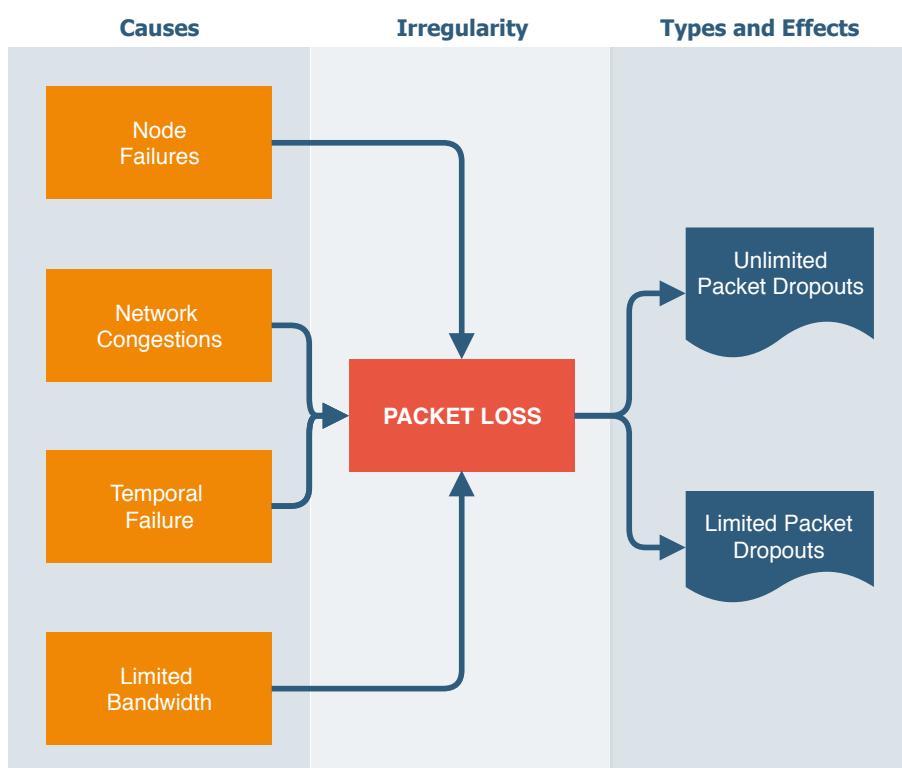


Figure 3.4: Packet loss diagram, showing the main causes (in orange) and effects (in dark blue) of time delay irregularity.

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

$$y(k) = \xi(k)z(k) + (1 - \xi(k))y(k-1), \quad (3.2)$$

where $z(k) \in \mathbb{R}^m$ is the k -th measured output transmitted to the estimator, $y(k) \in \mathbb{R}^m$ is the k -th measurement received by the estimator and $\xi(k) \sim Ber(p)$ is a Bernoulli stochastic process 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

$$\begin{aligned} y(k) = & \xi(k)z(k) + (1 - \xi(k))\xi(k-1)z(k-1) + (1 - \xi(k))(1 - \xi(k-1))\xi(k-2)z(k-2) + \dots \\ & + (1 - \xi(k))(1 - \xi(k-1))\dots(1 - \xi(k-N+1))\xi(k-N)z(k-N), \quad N \geq 1. \end{aligned} \quad (3.3)$$

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 .

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 associated with uncertain observations are shown in Figure 3.5.

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

$$y(k) = \xi(k)z(k) + v(k) \quad (3.4)$$

where $z(k) \in \mathbb{R}^m$ is the k -th measured output transmitted to the estimator, $y(k) \in \mathbb{R}^m$ is the k -th measurement received by the estimator and $\xi(k) \sim Ber(p(k))$ is a Bernoulli random variables, taking values of 0 or 1, with probabilities $p(k)$ and $1 - p(k)$, respectively.

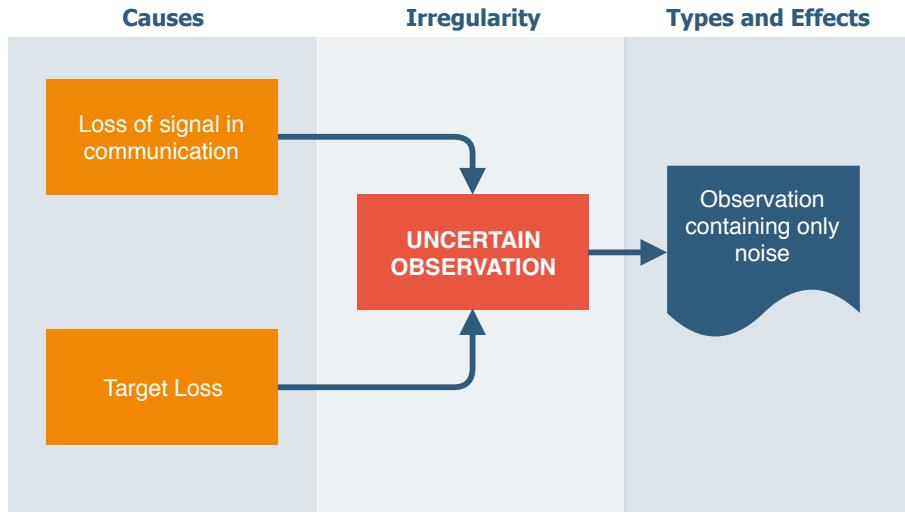


Figure 3.5: Uncertain observations diagram, showing the main causes (in orange) and effects (in dark blue) of time delay irregularity.

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 the estimator, making it sub-optimal. (Nahi, 1969) developed an optimal recursive estimator that uses the information of the stochastic process $\xi(k)$ 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 network-induced issues, such as irregular faults on samplers, oscillated loads, intermittent saturation or even variations in system components or

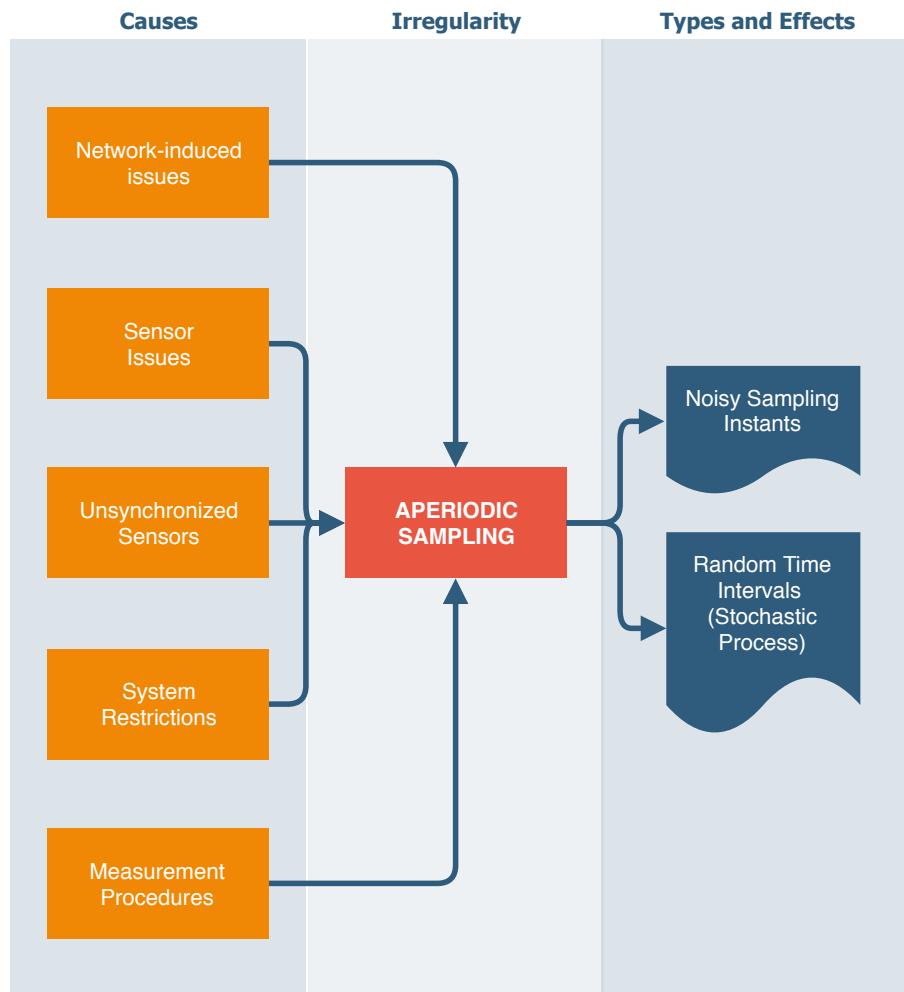


Figure 3.6: Aperiodic sampling diagram, showing the main causes (in orange) and types (in dark blue) of irregularity.

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, 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 geo-

graphical 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 variations 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 replaced in favor of an event-based scheme. In such strategy, an event-triggering mechanism is responsible for determining the sampling instants, as depicted in 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, lower-bounds can be defined both for the Δ -value or for some explicit minimum inter-event time.

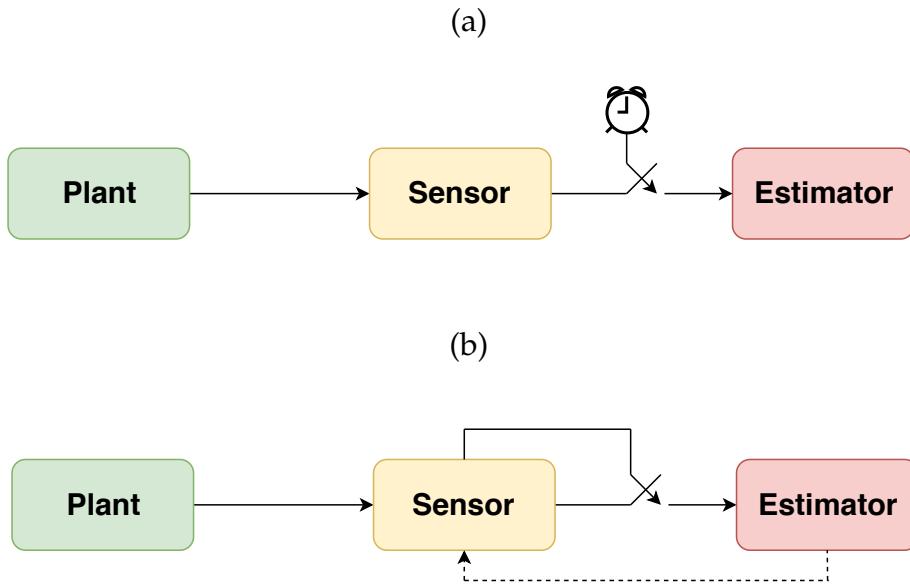


Figure 3.7: (a) Time-driven and (b) event-driven sampling schemes. The connection between sensor and estimator is triggered by different mechanisms.

The measurement model with aperiodic sampling can be defined as

$$y(t_k) = z(t_k) + v(t_k) \quad (3.5)$$

where t_k is the random sampling time instant and $z(t_k)$ is the transmitted measurement.

Generalizations of aperiodic sampling can be divided in two categories, based on how the estimator perceives the irregularity: as time signal noise added to a periodic pattern; or as a stochastic process, as illustrated in Figure 3.8.

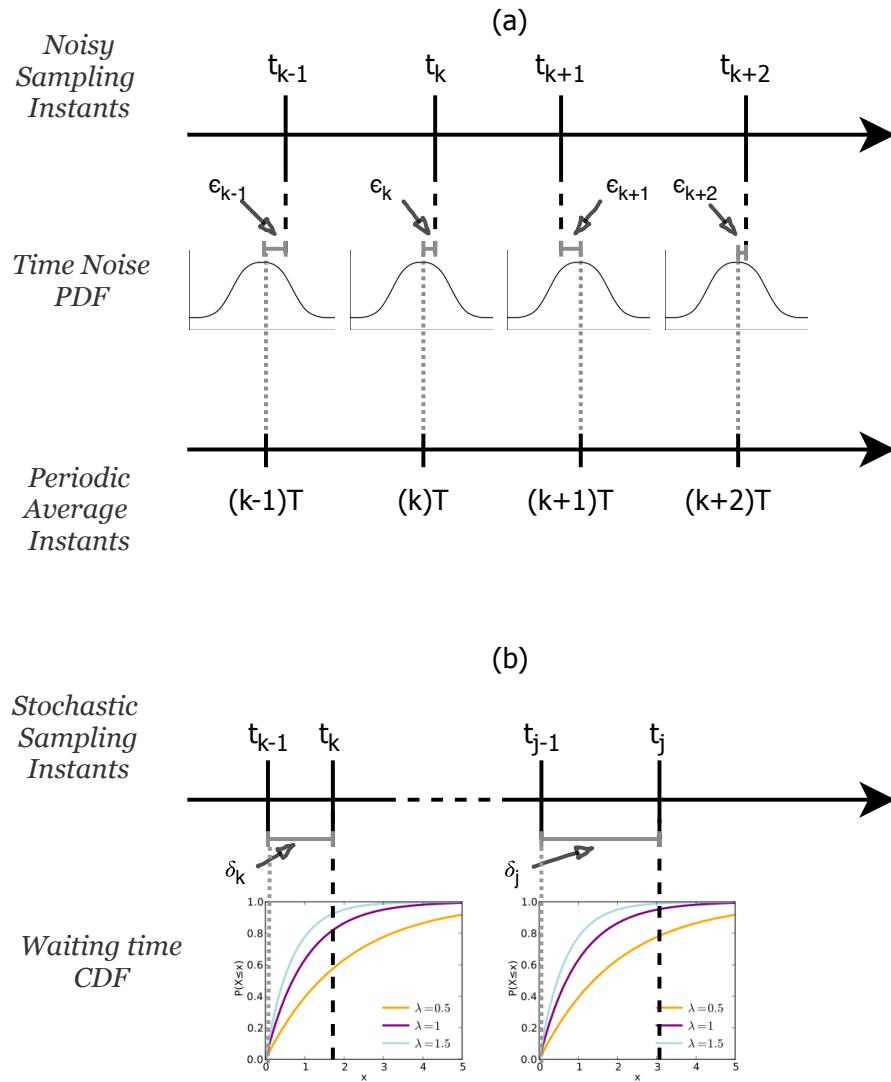


Figure 3.8: Two models of aperiodic sampling: (a) noisy sampling over periodic intervals, with a Gaussian random variable added to expected time instants $k\tau$; and (b) sampling instants modeled as a stochastic process, with time intervals characterized by an exponential random variable. For (b), the 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:

$$\begin{aligned} t_k &\triangleq kT + \epsilon_k, \\ \delta_k &\triangleq t_k - t_{k-1} \end{aligned} \tag{3.6}$$

where t_k is the k^{th} sampling instant, T is the constant 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 $t_k \sim \mathcal{N}(kT, \sigma^2)$, $\forall k \in \mathbb{N}$, 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:

$$\begin{aligned} \delta_k &\triangleq t_k - t_{k-1}, \\ \delta_0 &\triangleq t_0 \end{aligned} \tag{3.7}$$

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 done in ([Kanchanaharuthai and Wongsaisuwan, 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 proved 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 δ_k is $E[\delta_k] = 1/\lambda$ and its variance is $Var[\delta_k] = 1/\lambda^2$.

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 effects associated with multi-rate sampling irregularity.

Many industrial processes need to control variables that can be measured by online

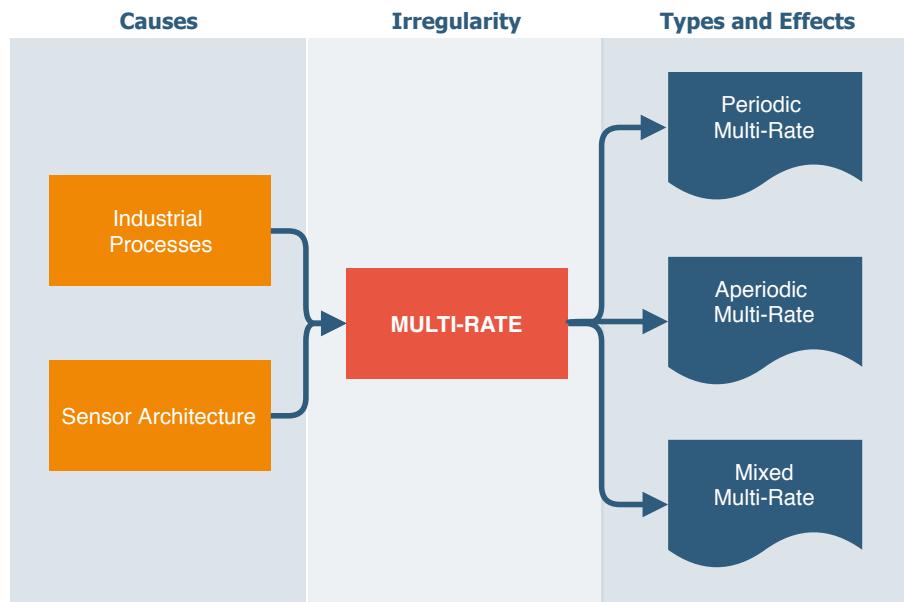


Figure 3.9: Multi-rate sampling diagram, showing the main causes (in orange) and types (in dark blue) of multi-rate sampling.

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](#). Periodic multi-rate measurements can be modeled as

$$y_i(k_i T_i) = z(k_i T_i) + v_i(k_i T_i) \quad (3.8)$$

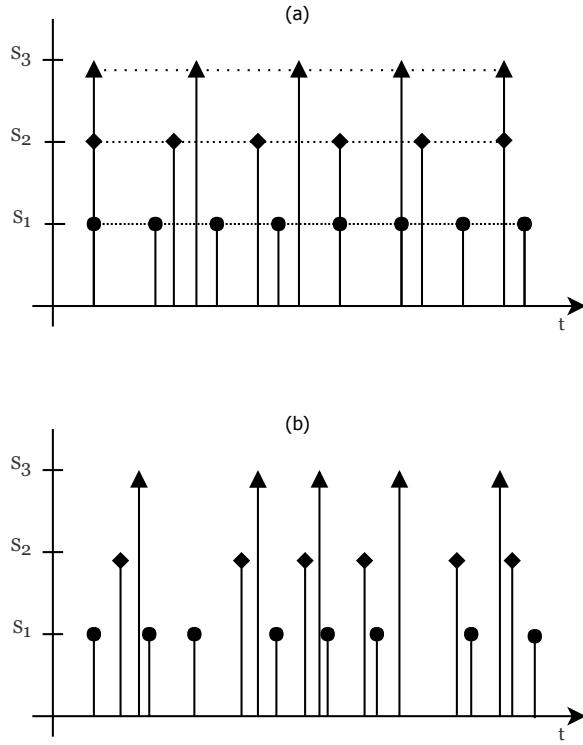


Figure 3.10: (a) Periodic and (b) aperiodic multi-rate sampling scheme. Labels \$S_1, S_2, S_3\$ refer to the sampling instants of three different sensors, from the highest sampling rate to the lowest sampling rate.

where $y_i(k_i T_i)$ represents the k_i^{th} observation from sensor i with periodic sampling rate T_i and $z(k_i T_i)$ is the k -th transmitted measurement of the i -th 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 is able to use the information on the time that measurements were taken, there are many algorithms that provide interesting results.

In general, multi-sensor data fusion techniques for state estimation require that exact measurement timestamps 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 timestamps 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 being time stamped at the moment it was taken, or when they

are time stamped using temporal information from 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 timestamp (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 timestamps, with the assumption that they could be statistically characterized. They proposed an implementation of the covariance union (CU) technique, using the timestamp 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.

Table 3.1: Comparison of NEES consistency test results for fusion of time delayed measurements with uncertain timestamps 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

As expected, the method with known delay was the most consistent, that is with $E[\text{NEES}]$ results closer to 2. CU and PDAF were both consistent, with slightly overestimated state error. They argued that, although PDAF obtained better consistency 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 timestamp leads to considerable performance degradation.

Another approach to overcome unknown timestamps 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 works of (Sivrikaya and Yener, 2004) and (Römer, 2003) provide detailed reviews on the time synchronization problem in sensor networks.

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).

Another 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.

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 requires 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 timestamps are not available.

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	$29.1 \mu s$ 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	$16.9 \mu s$ High E. E. Low Comp.
DMTS	Reduces the number of message exchanges	Restricted to low resolution and low frequency external clocks	$32 \mu s$ 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	$945 \mu s$ 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	$1.48 \mu s$ High E. E. High Comp.
LEETS	Low power consumption and low amount of message exchanges	Requires a GPS receiver in the root node	$30 \mu s$ High E. E. Low Comp.
TDP	Tolerant to message losses, high mobility and performs well even without external servers	Very high convergence time	$100 \mu s$ High E. E. High Comp.
TSST	Low synchronization error	Not scalable	Unknown Low E. E. Low Comp.

State Estimation for Sampled-Data Systems

In this chapter, the investigation in the field of sensor fusion under irregular sampling is narrowed down 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 unscented transform is also explored. In Section 4.3 we describe the peculiarities of the filtering algorithms for when the correct timestamp 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 Model 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 reduce computational cost in 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 is formulated as a discrete-time model. First, we reproduce the stochastic nonlinear sampled system models as follows

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

$$y(t_k) = g(x(t_k), v(t_k), t_k), \quad (4.2)$$

where $f: \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}^+ \rightarrow \mathbb{R}^n$ and $g: \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^+ \rightarrow \mathbb{R}^m$ 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

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

$$t_{k+1} = t_k + h_k, \quad (4.4)$$

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

$$k_1 = h_k f(t_k, x(t_k)), \quad (4.5)$$

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

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

$$k_4 = h_k f(t_k + h_k, x(t_k) + k_3), \quad (4.8)$$

where $f(x(t), u(t), w(t), t)$ was represented as $f(t, x(t))$ 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 zero mean additive noise. Therefore, the discrete-time representation for the nonlinear sampled system becomes

$$x(t_{k+1}) = f_d(x(t_k), u(t_k), w(t_k), t_k), \quad (4.9)$$

$$y(t_k) = g(x(t_k), v(t_k), t_k), \quad (4.10)$$

where $f_d(\cdot)$ 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), the sampled system is given by

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

$$y(t_k) = Cx(t_k) + v(t_k), \quad (4.12)$$

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)

$$x(t_{k+1}) = A_d(t_k, t_{k+1})x(t_k) + B_d(t_k, t_{k+1})u(t_k) + w_d(t_k, t_{k+1}), \quad (4.13)$$

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

$$A_d(t_k, t_{k+1}) = e^{A(t_{k+1}-t_k)}, \quad (4.14)$$

$$B_d(t_k, t_{k+1}) = \left(\int_{\tau=0}^{t_{k+1}-t_k} e^{A\tau} \delta\tau \right) B, \quad (4.15)$$

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

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

$$Q_d(t_k, t_{k+1}) = \int_{\tau=0}^{t_{k+1}-t_k} e^{A\tau} G Q G^T e^{A^T \tau} d\tau, \quad (4.17)$$

For state estimation purpose, we are interested in the calculation of (4.17) 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 (4.17) 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

$$x(t_{k+1}) = A_d(t_k, t_{k+1})x(t_k) + B_d(t_k, t_{k+1})u(t_k) + G_d(t_k, t_{k+1})w(t_k), \quad (4.18)$$

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 $G_d(t_k, t_{k+1}) = \left(\int_{\tau=0}^{t_{k+1}-t_k} e^{A\tau} \delta\tau \right) G$ and $Q_d(t_k, t_{k+1}) = (t_{k+1} - t_k)Q$.

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 , 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

$$SNR = \frac{P_{\text{signal}}}{P_{\text{noise}}}, \quad (4.19)$$

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

$$SNR = \frac{\sigma_{\text{signal}}^2}{\sigma_{\text{noise}}^2}. \quad (4.20)$$

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

$$SNR_{dB} = 10 \log_{10} \frac{P_{\text{signal}}}{P_{\text{noise}}}. \quad (4.21)$$

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 prior information. For that, both the states and the arrived information are modeled as stochastic processes (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 $y_1, \dots, y_k \in \mathbb{R}^m$ and input vector sequence $u_1, \dots, u_k \in \mathbb{R}^p$. The corresponding conditional PDF, $\rho(x_k | (y_1, \dots, y_k))$, 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 $\rho(x_k | (y_1, \dots, y_k))$ 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)

$$\hat{x}_k^{MAP} \triangleq \arg \max_{x_k} \hat{\rho}(x_k | (y_1, \dots, y_k)), \quad (4.22)$$

$$\hat{x}_k^{MMSE} \triangleq \arg \min_{x_k} E[(\hat{x}_k - x_k)^T (\hat{x}_k - x_k) | (y_1, \dots, y_k)], \quad (4.23)$$

where \hat{x}_k is the estimated value of $x_k \in \mathbb{R}^n$, $\hat{\rho}(x_k | (y_1, \dots, y_k))$ is the estimated posterior PDF of x_k given the observation sequence, and $E[(\hat{x}_k - x_k)^T (\hat{x}_k - x_k) | (y_1, \dots, y_k)]$ 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* $\hat{\rho}(x_k|y_1, \dots, y_k)$ 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 to a Markov process is presented in Proposition 4.2.1. But first, we present two facts related to Markov processes .

Fact 4.2.1. (Papoulis, 1984) 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

$$\rho(x_{k+1}|(x_0, \dots, x_k)) = \rho(x_{k+1}|x_k), \quad (4.24)$$

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

$$\rho(y_k|(x_0, \dots, x_k, y_0, \dots, y_{k-1})) = \rho(y_k|x_k). \quad (4.25)$$

□

Fact 4.2.2. (Papoulis, 1984) 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

$$\rho(x_n|x_s) = \int_{-\infty}^{\infty} \rho(x_n|x_r) \rho(x_r|x_s) dx_r. \quad (4.26)$$

□

Note that we assume 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 (4.24) and (4.25), respectively.

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

$$\rho(x_k|(y_1, \dots, y_{k-1})) = \int_{\mathbb{R}^n} \rho(x_k|x_{k-1}) \rho(x_{k-1}|y_1, \dots, y_{k-1}) dx_{k-1}, \quad (4.27)$$

$$\rho(x_k|(y_1, \dots, y_k)) = \frac{\rho(y_k|x_k) \rho(x_k|(y_1, \dots, y_{k-1}))}{\rho(y_k|(y_1, \dots, y_{k-1}))} \quad (4.28)$$

where $k \in \mathbb{N}$, $\rho(y_k|x_k)$ is the *likelihood PDF*, $\rho(x_k|(y_1, \dots, y_{k-1}))$ 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 $\rho(y_k|(y_1, \dots, y_{k-1}))$ is the *evidence*, also referred to as normalizing factor, or marginalization.

□

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

$$\rho(x_k|(y_1, \dots, y_k)) = \frac{\rho((y_1, \dots, y_k)|x_k) \rho(x_k)}{\rho(y_1, \dots, y_k)}. \quad (4.29)$$

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

$$\rho((a_1, \dots, a_k)|(a_{k+1}, \dots, a_n)) = \frac{\rho(a_1, \dots, a_k, \dots, a_n)}{\rho(a_{k+1}, \dots, a_n)}, \quad (4.30)$$

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

$$\rho(a_1, \dots, a_n) = \rho(a_n|a_1, \dots, a_{n-1}) \rho(a_1, \dots, a_{n-1}), \quad (4.31)$$

it is possible to rewrite (4.29) as

$$\rho(x_k|(y_1, \dots, y_k)) = \frac{\rho(y_k|(y_1, \dots, y_{k-1}, x_k)) \rho(y_1, \dots, y_{k-1}|x_k) \rho(x_k)}{\rho(y_k|(y_1, \dots, y_{k-1})) \rho(y_1, \dots, y_{k-1})}. \quad (4.32)$$

From Fact 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

$$\rho((y_1, \dots, y_{k-1})|x_k) = \frac{\rho(x_k|(y_1, \dots, y_{k-1})) \rho((y_1, \dots, y_{k-1}))}{\rho(x_k)}. \quad (4.33)$$

Finally, by combining all together, we have

$$\rho(x_k|(y_1, \dots, y_k)) = \frac{\rho(y_k|x_k) \rho(x_k|(y_1, \dots, y_{k-1})) \rho((y_1, \dots, y_{k-1})) \rho(x_k)}{\rho(y_k|(y_1, \dots, y_{k-1})) \rho(y_1, \dots, y_{k-1}) \rho(x_k)}, \quad (4.34)$$

which, after canceling the equal terms (in gray), proves (4.28).

To prove (4.27), we introduce a predicted state x_{k+1} in the posterior PDF, that is $\rho(x_{k+1}, x_k | (y_1, \dots, y_k))$ (Bergman, 1999). Rewriting the new conditional PDF with the aid of (4.30), (4.31) and Fact 4.2.1, we have

$$\begin{aligned}\rho(x_{k+1}, x_k | (y_1, \dots, y_k)) &= \rho(x_{k+1} | (x_k, y_1, \dots, y_k)) \rho(x_k | (y_1, \dots, y_k)) \\ &= \rho(x_{k+1} | (x_k)) \rho(x_k | (y_1, \dots, y_k)).\end{aligned}\quad (4.35)$$

The integration of both sides of (4.35) with respect to x_k yields (4.27), which is also a Chapman-Kolmogorov equation. This last piece of proof is also an application of Fact 4.2.2.

□

The evidence term in the denominator of (4.28) 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 Fact 4.2.2

$$\begin{aligned}\rho(y_k | (y_1, \dots, y_{k-1})) &= \int_{\mathbb{R}^n} \rho(y_k | x_k) \rho(x_k | (y_1, \dots, y_{k-1})) dx_k, \\ &= z_k.\end{aligned}\quad (4.36)$$

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 (4.27), in the *data assimilation* step, given by (4.28). Thus, (4.28) 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

$$\begin{aligned}\rho(x_k | (y_1, \dots, y_{k-1})) &= \mathcal{N}(x_k | \hat{x}_p, \hat{\sigma}_p) \\ &= \frac{1}{\sqrt{2\pi}\hat{\sigma}_p} \exp\left(-\frac{(x_k - \hat{x}_p)^2}{2\hat{\sigma}_p^2}\right),\end{aligned}\quad (4.37)$$

$$\begin{aligned}\rho(y_k | x_k) &= \mathcal{N}(y_k | x_k, \hat{\sigma}_e) \\ &= \frac{1}{\sqrt{2\pi}\hat{\sigma}_e} \exp\left(-\frac{(y_k - x_k)^2}{2\hat{\sigma}_e^2}\right),\end{aligned}\quad (4.38)$$

where the subscript p stands for *prior* and the subscript e stands for *evidence*. The *prior* PDF defined in (4.37) is obtained by forecasting with the aid of a process model, and the *likelihood* PDF given by (4.38) is obtained by a linear observation model, say $y_k = x_k + w_k$, with $w_k \sim \mathcal{N}(0, \sigma_e)$. Under the MAP estimation method, we can combine (4.22) and (4.28), discarding the normalization factor, which is independent of x_k , yielding

$$\begin{aligned}\hat{x}_k^{MAP} &= \arg \max_{x_k} \rho(y_k | x_k) \rho(x_k | (y_1, \dots, y_{k-1})) \\ &= \arg \max_{x_k} \mathcal{N}(y_k | x_k, \hat{\sigma}_e) \mathcal{N}(x_k | \hat{x}_p, \hat{\sigma}_p),\end{aligned}\quad (4.39)$$

and from the multiplication of (4.37) and (4.38) we have that

$$\begin{aligned}\mathcal{N}(y_k | x_k, \hat{\sigma}_e) \mathcal{N}(x_k | \hat{x}_p, \hat{\sigma}_p) &= \frac{1}{\sqrt{2\pi}\hat{\sigma}_p} \exp\left(-\frac{(x_k - \hat{x}_p)^2}{2\hat{\sigma}_p^2}\right) \times \frac{1}{\sqrt{2\pi}\hat{\sigma}_e} \exp\left(-\frac{(y_k - x_k)^2}{2\hat{\sigma}_e^2}\right) \\ &= \frac{1}{2\pi\hat{\sigma}_p\hat{\sigma}_e} \exp\left[-\left(\frac{(y_k - x_k)^2}{2\hat{\sigma}_e^2} + \frac{(x_k - \hat{x}_p)^2}{2\hat{\sigma}_p^2}\right)\right],\end{aligned}\quad (4.40)$$

whose quadratic terms can be expanded and, apart from a constant term that will be handled by the normalization factor discarded in (4.39), rearranged into

$$\begin{aligned}\mathcal{N}(y_k | x_k, \hat{\sigma}_e) \mathcal{N}(x_k | \hat{x}_p, \hat{\sigma}_p) &= \mathcal{N}(x_k | \mu_{\text{fused}}(y_k), \hat{\sigma}_{\text{fused}}) \\ &= \frac{1}{\sqrt{2\pi}\hat{\sigma}_{\text{fused}}} \exp\left(-\frac{(x_k - \mu_{\text{fused}}(y_k))^2}{2\hat{\sigma}_{\text{fused}}^2}\right),\end{aligned}\quad (4.41)$$

where

$$\begin{aligned}\mu_{\text{fused}}(y_k) \triangleq \hat{x}_{\text{posterior}}(y_k) &= \frac{\hat{\sigma}_e^2}{\hat{\sigma}_p^2 + \hat{\sigma}_e^2} \hat{x}_p + \frac{\hat{\sigma}_p^2}{\hat{\sigma}_e^2 + \hat{\sigma}_p^2} y_k \\ &= \hat{x}_p + \frac{\hat{\sigma}_p^2}{\hat{\sigma}_e^2 + \hat{\sigma}_p^2} (y_k - \hat{x}_p),\end{aligned}\quad (4.42)$$

and

$$\begin{aligned}\hat{\sigma}_{\text{fused}} \triangleq \hat{\sigma}_{\text{posterior}} &= \frac{\hat{\sigma}_p^2 \hat{\sigma}_e^2}{\hat{\sigma}_p^2 + \hat{\sigma}_e^2} \\ &= \hat{\sigma}_p^2 - \frac{\hat{\sigma}_p^2}{\hat{\sigma}_e^2 + \hat{\sigma}_p^2} \hat{\sigma}_p^2.\end{aligned}\quad (4.43)$$

The derivation from (4.42) and (4.43) comes from the multiplication of two Gaussian PDFs. Combining the results of (4.42) and (4.43) with (4.39), we have that

$$\begin{aligned}\hat{x}_k^{MAP} &= \arg \max_{x_k} \mathcal{N}(x_k | \hat{x}_{\text{posterior}}(y_k), \hat{\sigma}_{\text{posterior}}) \\ &= \hat{x}_{\text{posterior}}(y_k).\end{aligned}\quad (4.44)$$

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 assumptions, that is linearity and Gaussianity.

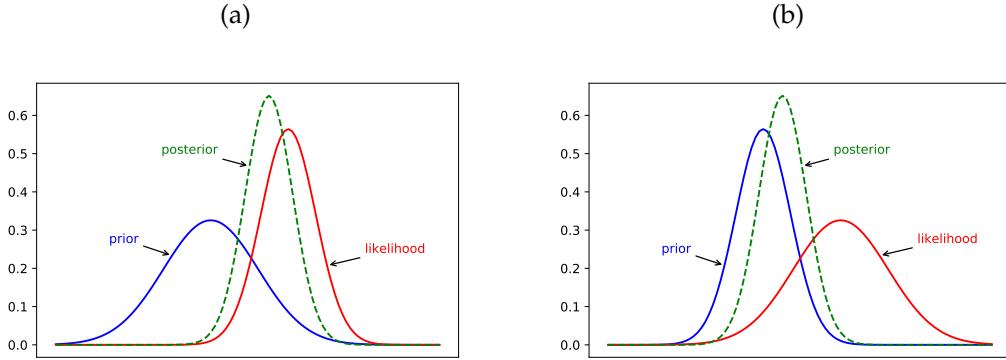


Figure 4.1: Posterior PDF obtained by the fusion of *prior* and *likelihood* densities. In (a) the variance of the *likelihood* is smaller than the variance of the *prior*, hence the *posterior* is closer to the *likelihood*. In (b) is the other way around.

Consider the Gauss-Markov discrete-time linear system

$$x_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + G_{k-1}w_{k-1}, \quad (4.45)$$

$$y_k = C_kx_k + v_k, \quad (4.46)$$

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-1} , 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

$$\mathcal{N}(x; \mu, P^{xx}) \triangleq \frac{1}{(2\pi)^{n/2} |P^{xx}|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^T (P^{xx})^{-1} (x - \mu)\right) \quad (4.47)$$

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,

$$\rho(x, y) = \mathcal{N}\left(\begin{pmatrix} x \\ y \end{pmatrix}; \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}, \begin{pmatrix} P^{xx} & P^{xy} \\ (P^{xy})^T & P^{yy} \end{pmatrix}\right) \quad (4.48)$$

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

$$\rho(x) = \mathcal{N}(x; \bar{x}, P^{xx}) \quad (4.49)$$

$$\rho(y) = \mathcal{N}(y; \bar{y}, P^{yy}) \quad (4.50)$$

$$\rho(x|y) = \mathcal{N}(x; \bar{x} + P^{xy}(P^{yy})^{-1}(y - \bar{y}), P^{xx} - P^{xy}(P^{yy})^{-1}(P^{xy})^T) \quad (4.51)$$

$$\rho(y|x) = \mathcal{N}(y; \bar{y} + (P^{xy})^T(P^{xx})^{-1}(x - \bar{x}), P^{yy} - (P^{xy})^T(P^{xx})^{-1}P^{xy}) \quad (4.52)$$

□

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

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

$$\rho(y|x) = \mathcal{N}(y; Ax, P^{yy}), \quad (4.54)$$

then, the joint and marginal PDFs are defined by

$$\rho(x,y) = \mathcal{N}\left(\begin{pmatrix} x \\ y \end{pmatrix}; \begin{pmatrix} \bar{x} \\ A\bar{x} \end{pmatrix}, \begin{pmatrix} P^{xx} & P^{xx}A^T \\ AP^{xx} & AP^{xx}A^T + P^{yy} \end{pmatrix}\right), \quad (4.55)$$

$$\rho(y) = \mathcal{N}(y; A\bar{x}, AP^{xx}A^T + P^{yy}) \quad (4.56)$$

□

From the Gauss-Markov assumptions, we can rewrite (4.45) and (4.46) from a probabilistic perspective, as

$$\rho(x_k|x_{k-1}) = \mathcal{N}(x_k; A_{k-1}\bar{x}_{k-1} + B_{k-1}u_{k-1}, Q_k), \quad (4.57)$$

$$\rho(y_k|x_k) = \mathcal{N}(y_k; C_k\bar{x}_k, R_k), \quad (4.58)$$

where (4.57) is the *transition density* representing the system dynamics and (4.58) 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

$$\rho(x_k | (y_1, \dots, y_{k-1})) = \mathcal{N}(x_k; \hat{x}_{k|k-1}, P_{k|k-1}^{xx}), \quad (4.59)$$

$$\rho(x_k | (y_1, \dots, y_k)) = \mathcal{N}(x_k; \hat{x}_{k|k}, P_{k|k}^{xx}), \quad (4.60)$$

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

$$\rho(x_{k-1} | (y_1, \dots, y_{k-1})) = \mathcal{N}(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1}^{xx}), \quad (4.61)$$

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

Now we combine the *forecast steps* from (4.27) and from (4.59), using the process model from (4.57), the previous estimate from (4.61) and the identities from Property 4.2.2, yielding

$$\begin{aligned} \rho(x_k | (y_1, \dots, y_{k-1})) &= \int_{\mathbb{R}^n} \rho(x_k | x_{k-1}) \rho(x_{k-1} | y_1, \dots, y_{k-1}) dx_{k-1}, \\ &= \mathcal{N}(x_k; A_{k-1} \hat{x}_{k-1|k-1} + B_{k-1} u_{k-1}, A_{k-1} P_{k-1|k-1}^{xx} A_{k-1} - 1^T + G_{k-1} Q G_{k-1}^T), \end{aligned} \quad (4.62)$$

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

$$\hat{x}_{k|k-1} = A_{k-1} \hat{x}_{k-1|k-1} + B_{k-1} u_{k-1}, \quad (4.63)$$

$$P_{k|k-1}^{xx} = A_{k-1} P_{k-1|k-1}^{xx} A_{k-1}^T + G_{k-1} Q G_{k-1}^T. \quad (4.64)$$

And for the *data assimilation step*, we find the joint density of y_k with the *forecast* estimate from (4.62), using Property 4.2.1

$$\rho(x_k, y_k | (y_1, \dots, y_{k-1})) = \mathcal{N}\left(\begin{pmatrix} x_k \\ y_k \end{pmatrix}; \begin{pmatrix} \hat{x}_{k|k-1} \\ C_k \hat{x}_{k|k-1} \end{pmatrix}, \begin{pmatrix} P_{k|k-1}^{xx} & P_{k|k-1}^{xx} C_k^T \\ C_k P_{k|k-1}^{xx} & C_k P_{k|k-1}^{xx} C_k^T + R_k \end{pmatrix}\right), \quad (4.65)$$

and the marginal density for x_k is

$$\rho(x_k | (y_1, \dots, y_k)) = \mathcal{N}(x_k; \hat{x}_{k|k-1} + K_k (y_k - C_k \hat{x}_{k|k-1}), P_{k|k-1}^{xx} - P_{k|k-1}^{xx} C_k^T K_k^{-1} P_{k|k-1}^{xx} C_k^T), \quad (4.66)$$

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

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C_k \hat{x}_{k|k-1}), \quad (4.67)$$

$$P_{k|k}^{xx} = P_{k|k-1}^{xx} - P_{k|k-1}^{xx} C_k^T K_k^{-1} P_{k|k-1}^{xx} C_k^T. \quad (4.68)$$

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

$$K_k = P_{k|k-1}^{xx} C_k^T (C_k P_{k|k-1}^{xx} C_k^T + R_k)^{-1}. \quad (4.69)$$

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

$$\hat{y}_{k|k-1} \triangleq C_k \hat{x}_{k|k-1}, \quad (4.70)$$

$$P_{k|k-1}^{yy} \triangleq E[(y_k - \hat{y}_{k|k-1})(y_k - \hat{y}_{k|k-1})^T] = C_k P_{k|k-1}^{xx} C_k^T + R_k, \quad (4.71)$$

$$P_{k|k-1}^{xy} \triangleq E[(x_k - \hat{x}_{k|k-1})(y_k - \hat{y}_{k|k-1})^T] = P_{k|k-1}^{xx} C_k^T, \quad (4.72)$$

$$(4.73)$$

we can simplify *data assimilation* step given by (4.67), (4.68) and (4.69) as

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}), \quad (4.74)$$

$$P_{k|k}^{xx} = P_{k|k-1}^{xx} - P_{k|k-1}^{yy} K_k^{-1} (P_{k|k-1}^{yy})^T, \quad (4.75)$$

$$K_k = P_{k|k-1}^{xy} (P_{k|k-1}^{yy})^{-1}. \quad (4.76)$$

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 $P_{k|k-1}^{xx}$, using (4.63) and (4.64). Calculate Kalman gain K_k by (4.69).

Data assimilation step: Using the measurement vector y_k and Kalman gain, update estimations from previous step with (4.67) and (4.68), obtaining estimates $\hat{x}_{k|k}$ and $P_{k|k}^{xx}$.

□

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 a Monte Carlo approach.

First, we define a set of samples $\chi \triangleq [\chi_1, \dots, \chi_{2n}] \in \mathbb{R}^{n \times 2n}$, 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

$$\sum_{i=0}^{2n} \gamma_i \chi_i = \bar{x}, \quad (4.77)$$

$$\sum_{i=0}^{2n} \gamma_i [(\chi_i - \bar{x})(\chi_i - \bar{x})^T] = \bar{P}^{xx}, \quad (4.78)$$

$$\sum_{i=0}^{2n} \gamma_i = 1. \quad (4.79)$$

where $\gamma_i, \forall i > 1$ are called weights. There are various ways of determining the sigma points ([Menegaz et al., 2015](#)), 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 $\chi = [\chi_1 \ \chi_2 \ \dots \ \chi_{2n}]$ given by

$$\chi \triangleq \begin{bmatrix} \bar{x}\mathbf{1}_{1 \times 2n} + \sqrt{n}(\bar{P}^{xx})^{1/2} & \bar{x}\mathbf{1}_{1 \times 2n} - \sqrt{n}(\bar{P}^{xx})^{1/2} \end{bmatrix}, \quad (4.80)$$

with weights

$$\gamma = \frac{1}{2n}. \quad (4.81)$$

Such simplification has one less tuning 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 relating two random variables y and x

$$y = f(x, d), \quad (4.82)$$

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 (4.80) and (4.81) respectively. Propagate each sigma point χ_i through the nonlinear transformation given by (4.82) to find estimated mean \hat{y} and covariance matrix P^{yy} of the RV y and the cross-covariance matrix P^{xy} , by

$$\Upsilon_i = f(\chi_i, d), \quad \forall i = 1, \dots, 2n \quad (4.83)$$

$$\hat{y} = \sum_{i=0}^{2n} \gamma_i \Upsilon_i, \quad (4.84)$$

$$P^{yy} = \sum_{i=0}^{2n} \gamma_i \left[(\Upsilon_i - \hat{y})(\Upsilon_i - \hat{y})^T \right], \quad (4.85)$$

$$P^{xy} = \sum_{i=0}^{2n} \gamma_i \left[(\Upsilon_i - \hat{y})(\chi_i - \bar{x})^T \right]. \quad (4.86)$$

For simplicity, we will refer to (4.83)-(4.86) as the function ψ_{UT} , given by

$$[\hat{y} \ P^{yy} \ P^{xy}] = \psi_{UT}(\bar{x}, P^{xx}, n, d, f) \quad (4.87)$$

□

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 $P_{k|k-1}^{xx}$ based on last estimation

$$\begin{bmatrix} \hat{x}_{k|k-1} & P_{k|k-1}^{xx} \end{bmatrix} = \psi_{UT} \left(\hat{x}_{k-1|k-1}, P_{k-1|k-1}^{xx}, n, d, f \right), \quad (4.88)$$

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 (4.88), compute the forecast observation estimate $\hat{y}_{k|k-1}$, observation vector covariance $P_{k|k-1}^{yy}$ and the cross-covariance between observation and state $P_{k|k-1}^{xy}$, via the UT

$$\begin{bmatrix} \hat{y}_{k|k-1} & P_{k|k-1}^{yy} & P_{k|k-1}^{xy} \end{bmatrix} = \psi_{UT} \left(\hat{x}_{k|k-1}, P_{k|k-1}^{xx}, n, d, g \right), \quad (4.89)$$

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 (4.74)-(4.76), reproduced for convenience

$$K_k = P_{k|k-1}^{xy} (P_{k|k-1}^{yy})^{-1}, \quad (4.90)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - \hat{y}_{k|k-1}), \quad (4.91)$$

$$P_{k|k}^{xx} = P_{k|k-1}^{xx} - P_{k|k-1}^{yy} K_k^{-1} (P_{k|k-1}^{yy})^T. \quad (4.92)$$

□

UKF algorithm addresses many of the drawbacks that appear in the EKF implementation, such as the necessity of differentiability and the performance loss due to systems that are poorly approximated by linearization. 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 does not depend

on 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 updated 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 at 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 .

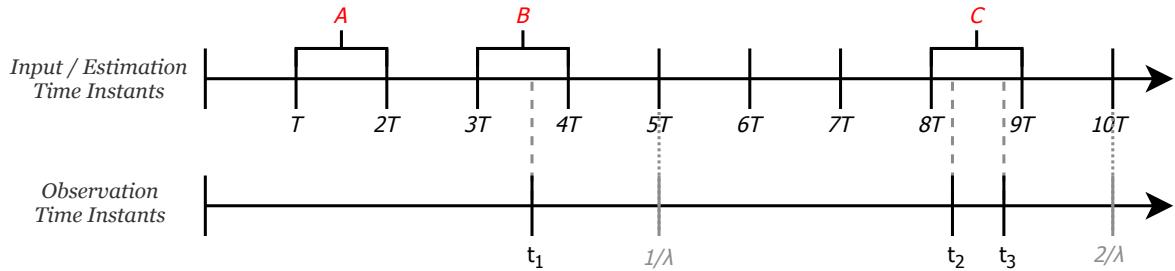


Figure 4.2: Input and estimation time instants labeled as iT and a realization of observation time instants, labeled as t_k . Expected time instants for observation is presented in gray, considering that $E[h_k] = \frac{1}{\lambda}$, and $\frac{1}{\lambda} = 5T$, that is $\alpha = 5$.

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 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 timestamp 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)

$$P(N(t) = n) = e^{-\lambda t} \frac{(\lambda t)^n}{n!}, \quad (4.93)$$

where λ is called rate or intensity parameter, $t = t_1 - t_0$ is the time difference between two time instants 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 inter-arrival time h_k , given by the distance between two consecutive arrivals, for the Poisson process has an exponential distribution, with a PDF given by

$$\rho_{h_k}(t) = \lambda e^{-\lambda t} \quad (4.94)$$

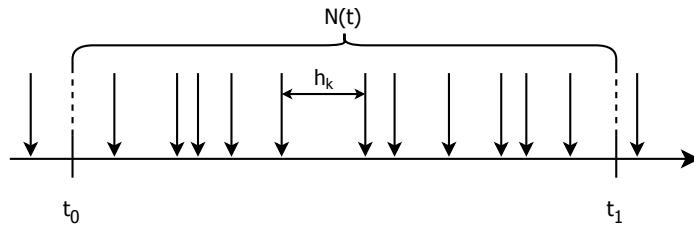


Figure 4.3: Arrivals of a Poisson process between a time interval given by $t = t_1 - t_0$. $N(t)$ is the RV representing the discrete amount of arrivals in the time interval and h_k is the distance between two consecutive arrivals.

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

$$y_i(nT + \xi_i) = Cx(nT + \xi_i) + v(nT + \xi_i), \quad n \in \mathbb{N}, i \in \{1, 2, \dots, N\}, \quad (4.95)$$

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.

Proposition 4.3.1. Let the phases of all sensors ξ_i , for $i = 1, 2, \dots, N$ 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 $h_N = \min(\xi_1, \xi_2, \dots, \xi_N)$ as the first interarrival time, occurring in $[0, N/\lambda]$. Then, the RV h_N converges in distribution to an exponential RV, that is

$$h_N \xrightarrow{D} \mathcal{E}(\lambda), \quad (4.96)$$

where \xrightarrow{D} means *converges in distribution* (Papoulis, 1984), as $N \rightarrow \infty$.

□

Proof. (Micheli and Jordan, 2002) 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) and the independence condition, yielding

$$\begin{aligned} \rho_{h_N}(\min_i(\xi_i) \leq t) &= 1 - \rho(\min_i(\xi_i) > t), \\ &= 1 - \rho(\xi_1 > t, \xi_2 > t, \dots, \xi_N > t), \\ &= 1 - \prod_{i=1}^N \rho(\xi_i > t), \\ &= 1 - \prod_{i=1}^N [1 - \rho(\xi_i \leq t)], \\ &= 1 - [1 - \rho(\xi_i \leq t)]^N. \end{aligned} \quad (4.97)$$

But since ξ_i for $i = 1, 2, \dots, N$ are i.i.d. uniform random variables in the interval $[0, N/\lambda]$, we can rewrite (4.97) as

$$\begin{aligned} \rho_{h_N}(\min_i(\xi_i) \leq t) &= 1 - \left(1 - \frac{\lambda t}{N}\right)^N, \\ &\rightarrow 1 - e^{-\lambda t}, \quad \text{as } N \rightarrow \infty, \end{aligned} \quad (4.98)$$

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

□

If we define $\lambda \triangleq N/T$, with a known time interval T and a sufficiently big amount of sensors N , 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

$$x(t_j^*) = f_d(x(t_{j-1}^*), u(t_{j-1}^*), w(t_{j-1}^*), t_{j-1}^*), \quad (4.99)$$

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 $t_0^* = 0$. In other words, 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. The input signals are considered to be held constant during each time interval, considering the last available information. The variable time interval δt_j^* is calculated according to the schematic presented by Figure 4.4.

We start with all counters set to 1. i represents the input counter, k the observation and j the estimation. When a signal is received, the algorithm checks if it derives from input or observation. In case it is an observation signal, δt_j^* is calculated by the difference of the measurement time instant t_k and the last estimation time instant t_{j-1}^* . Then a forecast step is performed, using input information from last input signal and finally the data observation data is assimilated, generating an estimate at instant t_j^* . Counters k and j are incremented and a new cycle begins. When input signal is received, we calculate the difference of the regular time instant iT from the last estimation time instant t_{j-1}^* . In this case, only forecast step is executed, using the received signal and the result is the estimate at instant t_j^* . i and j are incremented, and it starts all over again.

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

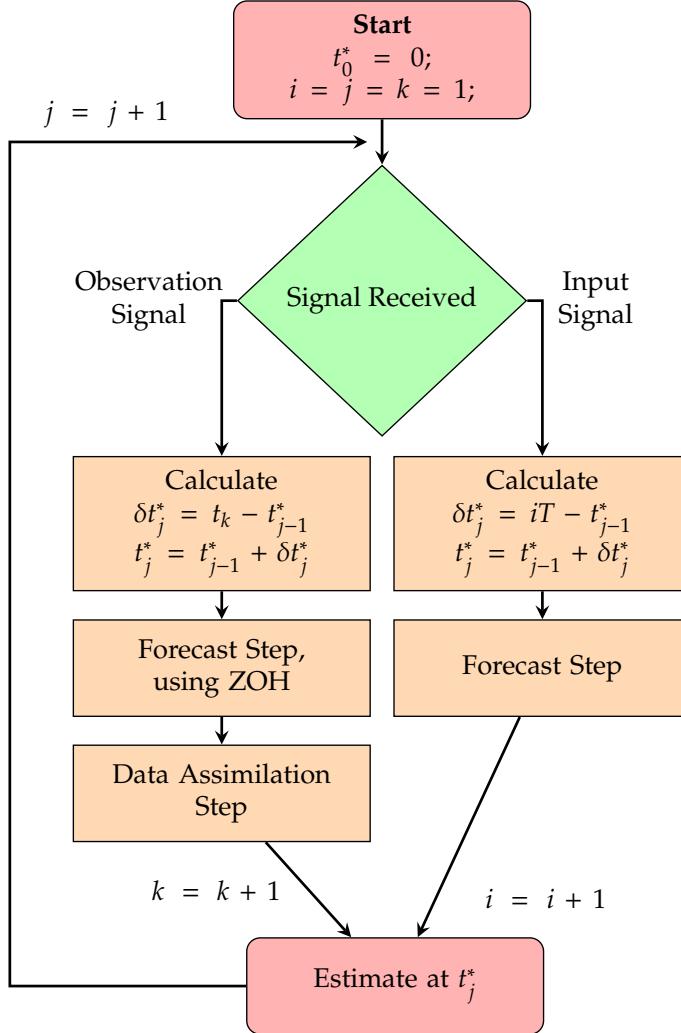


Figure 4.4: Illustrative schematic of the *online* estimator, with timestamp. indices i, j e k represent the input, estimation and observation signal counters, respectively.

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. 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 $\delta t_j^*, \forall j \geq 1$.

4.3.3 State Estimation Without Timestamp

For some applications, discretization at variable time intervals, as presented in Section 4.1, may be difficult to accomplish. In other situations measurements may be transmitted in a centralized manner, as discussed in Section 2.3, at regular time intervals, even though they were taken aperiodically like the models from Section 3.3.4. In such cases it is impossible to consider the time-stamp information in the estimation process. Thus process model (1.1) is discretized at a constant time interval, according to

$$x_n = f_d^*(x_{n-1}, u_{n-1}, w_{n-1}, n), \quad (4.100)$$

where $f_d^*(\cdot)$ represents the time-invariant discrete-time model 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 $9T$.

In practice, such configuration will impact the estimation by adding an error to the observation process (1.2). We illustrate this effect by an example.

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

$$G(s) = \frac{100}{s^2 + 2s + 100}, \quad (4.101)$$

whose continuous-time state space model can be represented by

$$\dot{x}(t) = \begin{bmatrix} -2 & -12.5 \\ 8 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 4 \\ 0 \end{bmatrix} u(t), \quad (4.102)$$

$$y(t) = [0 \quad 3.125] x(t). \quad (4.103)$$

A pseudo-random binary sequence (PRBS) with 200 samples is generated as input and oversampled by a factor of 100, resulting in 2×10^4 samples that we simulate over 2 seconds. The results are shown in Figure 4.5. We also 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 in Figure 4.6.

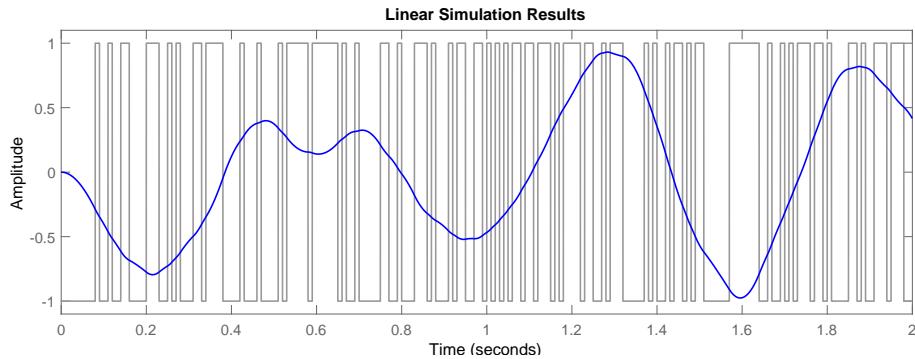


Figure 4.5: Linear simulation results for $G(s)$ in blue, considering the PRBS input in grey.

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

□

This approximation process leads to two sources of degradation. First, some information might be lost, if there are more than one observation 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.

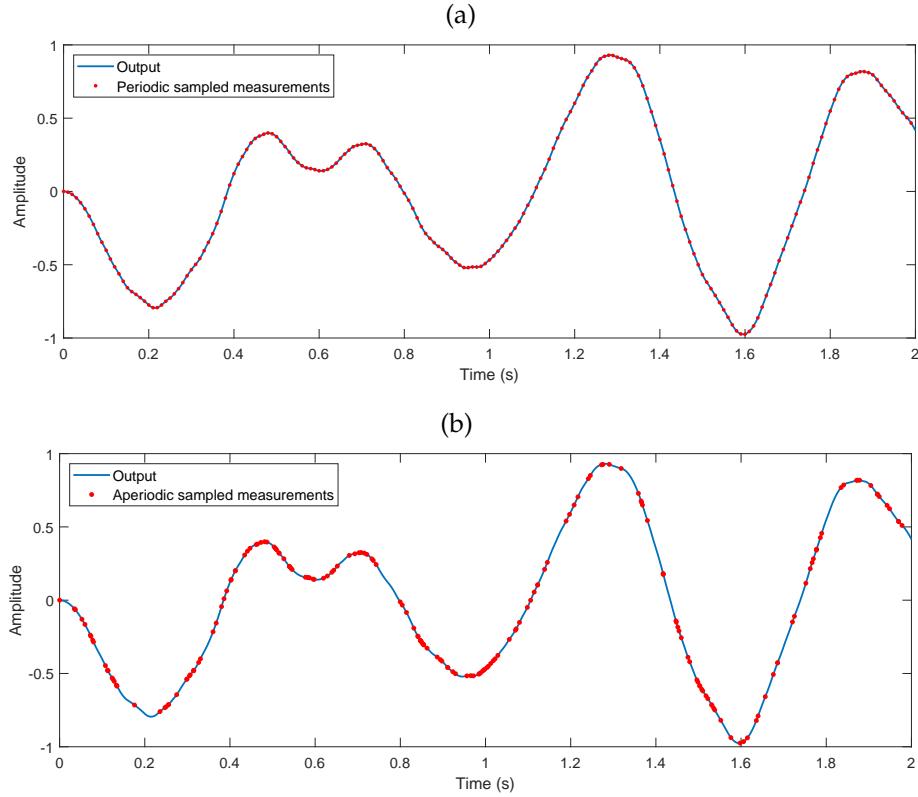


Figure 4.6: Periodic (a) and aperiodic (b) sampling of the output variable of $G(s)$. Aperiodic sampling instants sequence generated by a Poisson process with intensity $\lambda = 100$ Hz.

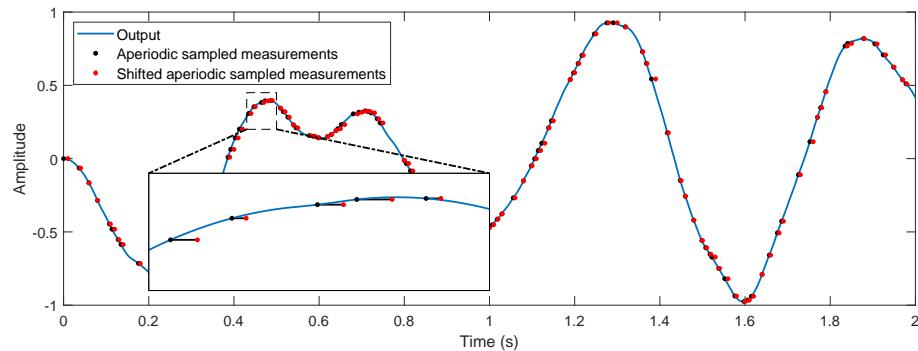


Figure 4.7: Effect of approximating the random time instants of the aperiodic sampling. Black dots are the true measurements and the red dots are the measurements shifted to the next multiple of $T = 0.001$.

4.4 Performance Metrics

In order to assess the performance degradation introduced by assimilating data at incorrect time instants, we need to define certain 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 $\hat{P}_{k|k}^{xx}$. For the linear case, the *posterior* conditional PDF $\rho(x_k|y_1, \dots, y_k)$ is Gaussian, according to (4.60), 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)

$$E[x_k - \hat{x}_{k|k}] \triangleq E[\tilde{x}_{k|k}] = 0, \quad (4.104)$$

$$E[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T] \triangleq E[\tilde{x}_{k|k}\tilde{x}_{k|k}^t] = P_{k|k}^{xx}, \quad (4.105)$$

where $\tilde{x}_{k|k}$ is the estimation error at time instant k . Condition (4.104) is called *unbiasedness* requirement, whereas (4.105) 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: (i) that state estimate errors are zero mean and compatible with the estimated state covariance; (ii) that innovations are also zero mean and compatible with their respective covariances; and (iii) that innovations are white. We adopt two tests proposed by (Bar-Shalom et al., 2001) 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

$$NEES_k \triangleq \tilde{x}_{k|k}^T (P_{k|k}^{xx})^{-1} \tilde{x}_{k|k}, \quad (4.106)$$

$$NIS_k \triangleq \eta_{k|k-1}^T (P_{k|k-1}^{yy})^{-1} \eta_{k|k-1}, \quad (4.107)$$

where $\eta_{k|k-1}$ is the innovation vector at time instant k and $P_{k|k-1}^{yy}$ its covariance matrix. 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 the

null hypothesis H_0 , meaning that both NEES and NIS at time instant kT is within a certain interval, considering the accepted region as

$$P\{NEES_k \in [r_1, r_2] | H_0\} = 1 - \alpha, \quad (4.108)$$

$$P\{NIS_k \in [r_1, r_2] | H_0\} = 1 - \alpha, \quad (4.109)$$

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

$$\left[\chi^2_2(0.025), \chi^2_2(0.975) \right] = [0.051, 7.38], \quad (4.110)$$

$$\left[\chi^2_4(0.025), \chi^2_4(0.975) \right] = [0.484, 11.1], \quad (4.111)$$

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^2_4 PDFs, with the 95% acceptance region.

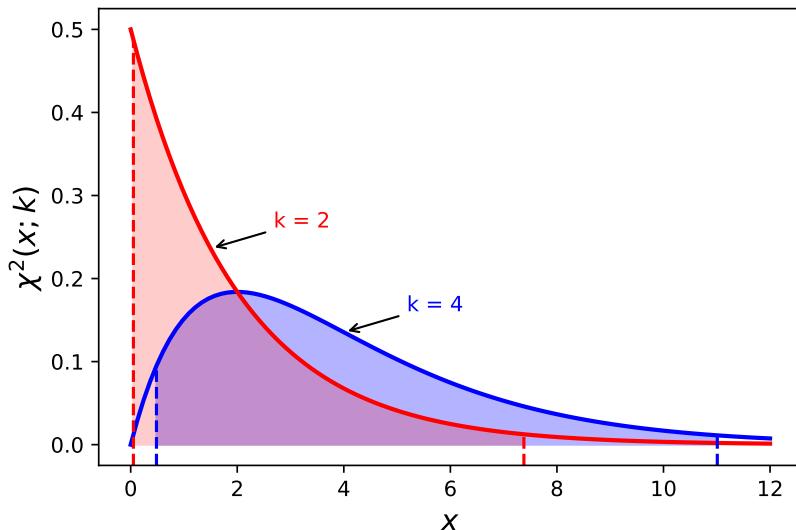


Figure 4.8: χ^2 PDFs for 2 (red) and 4 (blue) degrees of freedom. The vertical dashed lines represent the acceptance interval limits, for $\alpha = 5\%$, and the shaded area is the corresponding acceptance regions.

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

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (\hat{x}_{k|k} - x_k)^2}{N}} \quad (4.112)$$

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

Multiple runs are carried out and all accuracy and consistency data of all scenarios for both algorithms are stored. Then we construct box and whisker plots, so we can compare the results distributions. In order to assess performance degradation, we also conduct paired-samples t -tests to compare accuracy metrics of both algorithms. For that, each run considers the same conditions and the same random variables realizations for them both. The null hypothesis H_0 assumes that the true mean differences between metrics is zero, whereas the alternative hypothesis H_1 assumes it is not zero, that is

$$\begin{cases} H_0 : \mu_D = 0, \\ H_1 : \mu_D \neq 0, \end{cases} \quad (4.113)$$

where $\mu_D = RMSE_{w/o} - RMSE_w$ and suffixes w/o and w mean without and with timestamp, respectively. Since we use relatively high sample sizes, we expect differences with statistical significance for most of the results, considering a confidence level of 95%. However, for some scenarios, mean differences might not be practically significant, which has more meaning in the real world. Thus, we also calculate effect sizes to assess the difference between both algorithms, referred to as the d family of effect sizes. For that we use Cohen's d method for effect size estimation, given by

$$d = \frac{\overline{RMSE}_{w/o} - \overline{RMSE}_w}{s_D}, \quad (4.114)$$

where $\overline{RMSE}_{w/o}$ and \overline{RMSE}_w are the RMSE sample means of the algorithms without and with timestamps, respectively, and s_D the standard deviation of the paired-samples difference. This unitless measure provides a standardized statistic of the mean differences.

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 nonholonomic unicycle position system. Both systems are described in state-space representations. We design the simulation setup based on the investigation of the error resulting from shifting time instants of the measurements. Three simulation scenarios are chosen to enable the assessment 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 Introduced by Shifting Time Instants

In Example 4.3.1 we illustrated how measurements are assimilated by the state estimation algorithm when timestamps are not considered. A measurement error due to shifting time instants appears in the observation model, and it depends on the process model and the limits of the time shift. Considering the LTI system described by (4.11) and (4.12), such error is given by

$$e_k = C[x(t_k) - x(t_k + \delta_k)], \quad (5.1)$$

$$\delta_k = nT - t_k \quad (5.2)$$

where t_k is the most recent time instant in the interval $[(n-1)T, nT]$, $x(t) \in \mathbb{R}^n$ is the state

vector, $C \in \mathbb{R}^{m \times n}$ is the observation model matrix and δ_k is the shifted time interval. Dividing both sides of (5.1) by δ_k , yields

$$\frac{e_k}{\delta_k} = C \left(\frac{[x(t_k) - x(t_k + \delta_k)]}{\delta_k} \right), \quad (5.3)$$

$$e_k = C \left(\frac{[x(t_k) - x(t_k + \delta_k)]}{\delta_k} \right) \delta_k, \quad (5.4)$$

$$e_k \approx -C \frac{dx}{dt} \delta_k, \quad (5.5)$$

$$e_k \approx -\frac{dy}{dt} \delta_k. \quad (5.6)$$

Using the system from Example 4.3.1, we can observe some interesting aspects of the error e that is introduced by shifting the time instants of the measurements. In Figure 4.6 from Section 4.3.3 we observe that the error will always have negative values when the signal is increasing, and vice-versa, as expected. In Figure 5.1 we present the time series of such error overlapping the measurement, in different scales.

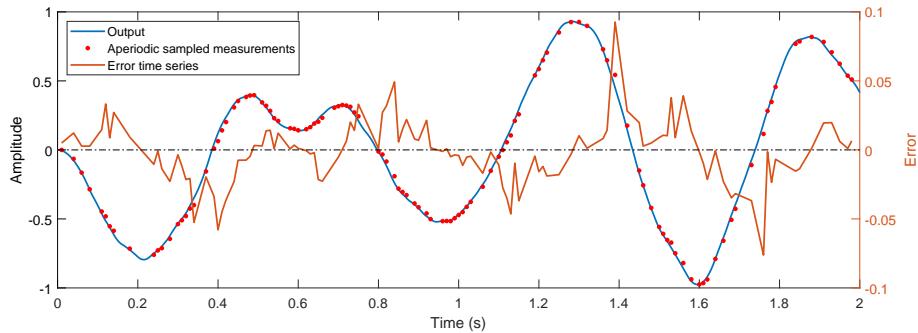


Figure 5.1: Added error time series (orange) as an effect of shifting time instants of the measurements (red) and output (blue). A gray dotted horizontal line is placed at the origin, for reference. Right scale (in orange) refers to the error data, whereas left scale (in black) refers to the output signal amplitude.

Additionally, the magnitude of the error is proportional to the opposite of the magnitude of the signal variation in the interval, that can be represented by its derivative. To better illustrate this aspect, 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 $G(s)$ (4.101) by s and simulating the output of the resulting system for the same input sequence. Figure 5.2 shows the original output, its derivative and the opposite of the error time series.

We choose to plot the opposite of the error, to get a better grasp of how the magnitudes are related. In fact, we observe that the error values are directly proportional to the opposite of 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 ([Wilk and Gnanadesikan, 1968](#)) shown in Figure 5.3 we observe significant deviations from the quantiles of the error data in comparison to those from the theoretical normal distribution. That was expected, since the error is signal-dependent as shown in (5.1).

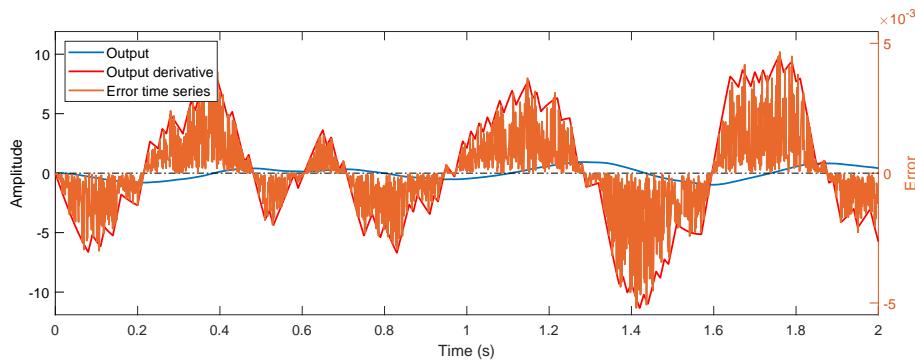


Figure 5.2: Opposite of the added error time series (orange) as an effect of shifting time instants of the measurements, output (blue) and its derivative (red). A gray dotted horizontal line is placed at the origin, for reference. We can see how the error is directly proportional to the opposite of the signal derivative.

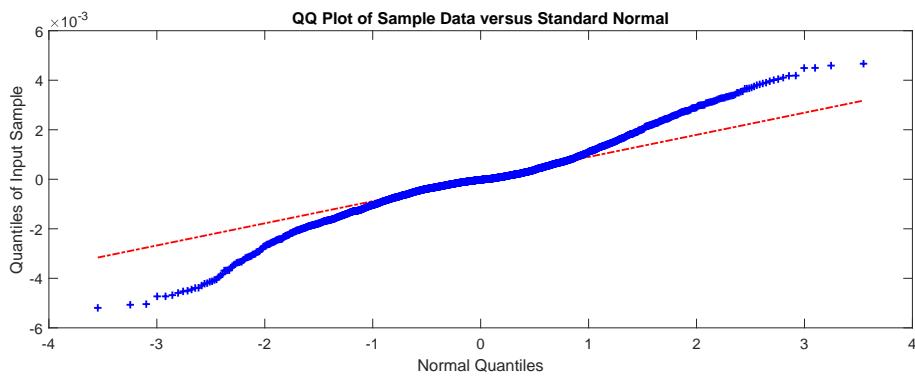


Figure 5.3: QQ-plot of the measurement error. Red line represents the expected quantiles of the standard normal distribution and the blue crosses, the error data quantiles.

The simulated example did not consider any signal corruption from the measurements. In real applications, however, there will be additive sensor noise, usually modeled as white and Gaussian. Depending on their SNR values, the effect of the error e might become irrelevant. To evaluate to what extent such claim is true, we carry out a

100-run simulation, varying both SNR and λ values for the system from Example 4.3.1. We estimate the RMSE contribution from e , by subtracting the total RMSE from the time-shifted measurements \tilde{y} from measurements RMSE y for each run. A surface plot of the results is shown in Figure 5.4. We can see that both parameters affect the RMSE contribution from e . Higher SNR values and lower λ are more sensible to the time shifting approximation for the signal. When measurement noise gets high enough, that is SNR lower than 60 dB, the RMSE contribution from e is very close to zero. Similarly, if the average sampling rate of the measurements are faster than 5 kHz, the noise levels are almost irrelevant, since the added RMSE is close to zero for all cases. These results motivate the design of different simulation scenarios for the state estimation problem.

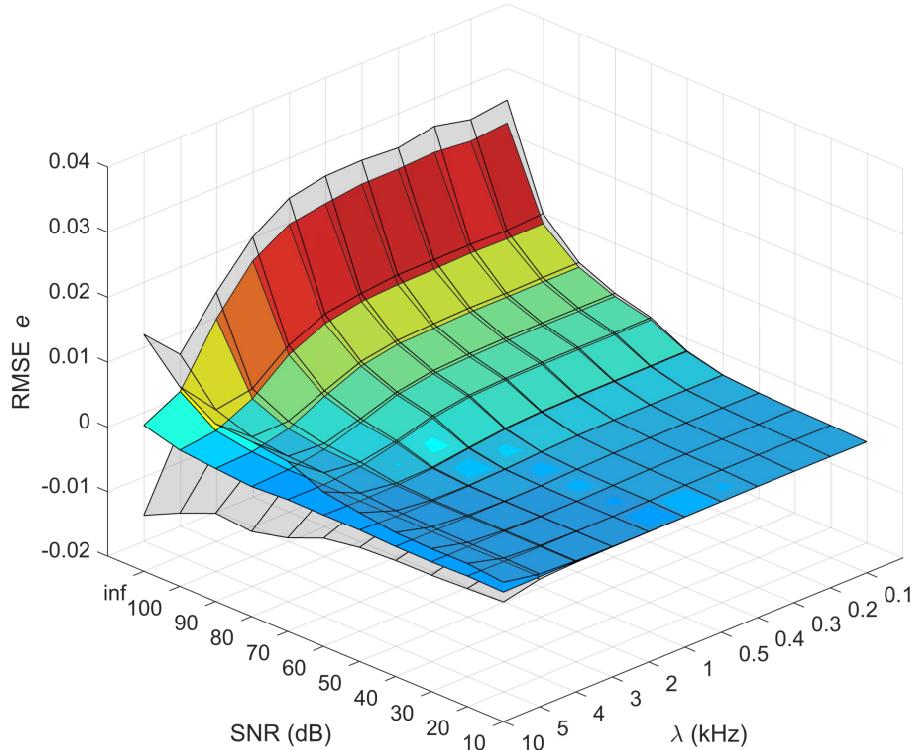


Figure 5.4: Surface plot of estimated RMSE from error e , as a function of SNR and λ . The transparent surfaces represent a confidence level of 95%.

5.2 Design of Simulation Setup

To evaluate performance and consistency effects of the aperiodic sampling in state estimation, we study three scenarios: (i) variation of noise level in signals, or SNR; (ii)

variation of average measurement sampling rate λ ; and (iii) 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 the performance of both algorithms, as discussed in the previous section. The second scenario assesses how the average sampling rate of measurements λ , keeping α constant, is related to the 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 frequency $1/T$ distances itself from the average observation sampling frequency λ . Additional errors due to measurement time instants approximation, as explained at the beginning of this chapter, are expected to have minor influence on the performance results.

Since estimation is performed in the presence of random quantities, we carry out several runs of simulations, 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 (4.112), and consistency tests $NEES$, given by (4.106) and (4.108), and NIS , given by (4.107) and (4.109). In Table 5.1 we review some of the concepts and symbols that are extensively used in this chapter.

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 easily grasped. Additionally, we choose the combination of two independent second order modes. One of the modes is a high-pass system with fast dynamics, and the other a low-pass system with slower dynamics. The idea is to enable the evaluation of the effects in a separate way for different frequency responses. That is motivated by the fact that the error introduced by the measurements temporal shifts is directly related to the signal derivative, as shown in Section 5.1.

Table 5.1: Review of concepts and symbols used in the numerical results

Symbol	Definition	Description
λ (Section 1.2)	$h_k \sim \mathcal{E}(\lambda)$	Average sampling rate of observations, in Hz, 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 (4.21)	$SNR_{dB} \triangleq 10 \log_{10} \frac{P_{signal}}{P_{noise}}$	Ratio between signal power P_{signal} and noise power P_{noise} in decibel, related to the level of data uncertainty
NEES (4.106)	$NEES \triangleq \tilde{x}^T (P^{xx})^{-1} \tilde{x}$	Value tested to be chi-squared distributed, in case of estimate error consistency
NIS (4.107)	$NIS \triangleq \eta^T (P^{yy})^{-1} \eta$	Value tested to be chi-squared distributed, in case of innovation consistency
μ_D (4.113)	$\mu_D \triangleq RMSE_{w/o} - RMSE_w$	Mean difference of RMSE from algorithms considering and not considering timestamp
Cohen's d (4.114)	$d \triangleq \frac{\overline{RMSE}_{w/o} - \overline{RMSE}_w}{s_D}$	Effect size estimate for the mean difference

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 5.5 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

$$G_{lp}(s) = \frac{100}{s^2 + 2s + 100}. \quad (5.7)$$

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:

$$G_{hp}(s) = \frac{s^2 - 0.001s}{s^2 + 200s + 10^6}. \quad (5.8)$$

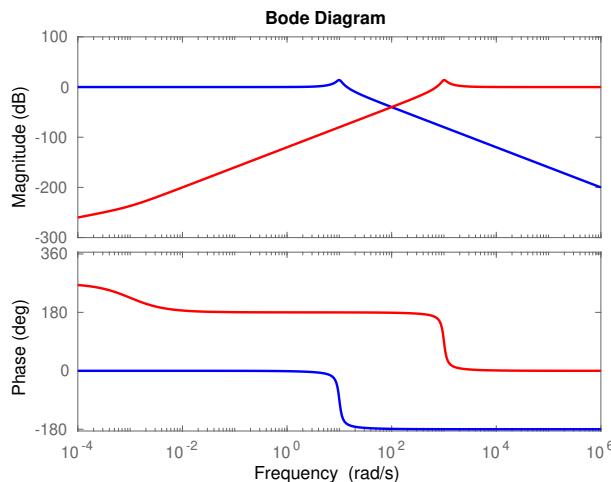


Figure 5.5: Bode diagram of both modes. Blue represent the low-pass system (5.7) and red the high-pass system (5.8).

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

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (5.9)$$

$$y(t) = Cx(t) + Du(t) \quad (5.10)$$

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

$$A = \begin{bmatrix} -100 & 994.99 & 0 & 0 \\ -994.99 & -100 & 0 & 0 \\ 0 & 0 & -1 & 9.949 \\ 0 & 0 & -9.949 & -1 \end{bmatrix}, \quad (5.11)$$

$$B = \begin{bmatrix} -24.6435 \\ -18.8943 \\ -4.1746 \\ -0.2675 \end{bmatrix}, \quad (5.12)$$

$$C = [24.41 \quad -21.2522 \quad -0.1537 \quad 2.3977], \quad (5.13)$$

$$D = 1. \quad (5.14)$$

Matrix A , given by (5.11), indicates two subsystem dynamically decoupled. The upper diagonal block refers to the high-pass system (5.8), whereas the lower diagonal block represents the low-pass system (5.7).

We simulate a pseudo-random binary sequence (PRBS) as input, with 200 samples, as shown in Figure 5.6. 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 5.7. The 20 seconds simulation for the high-pass system achieve steady state values very quickly, such that it is not easy to observe the influence of the system dynamics on the results. Thus, we choose the 2 seconds time frame to simulate the fourth order serialized systems, given by (5.9)-(5.14). The result of the linear simulation of the fourth order system is presented in Figure 5.8.

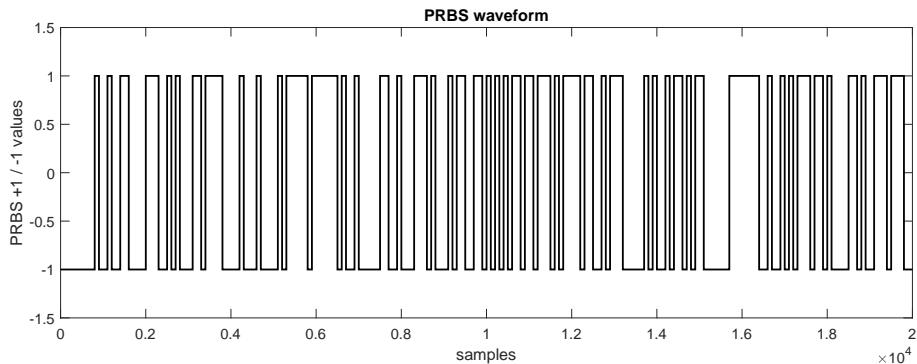


Figure 5.6: 200 PRBS samples used as input to simulate system (5.9)-(5.14)

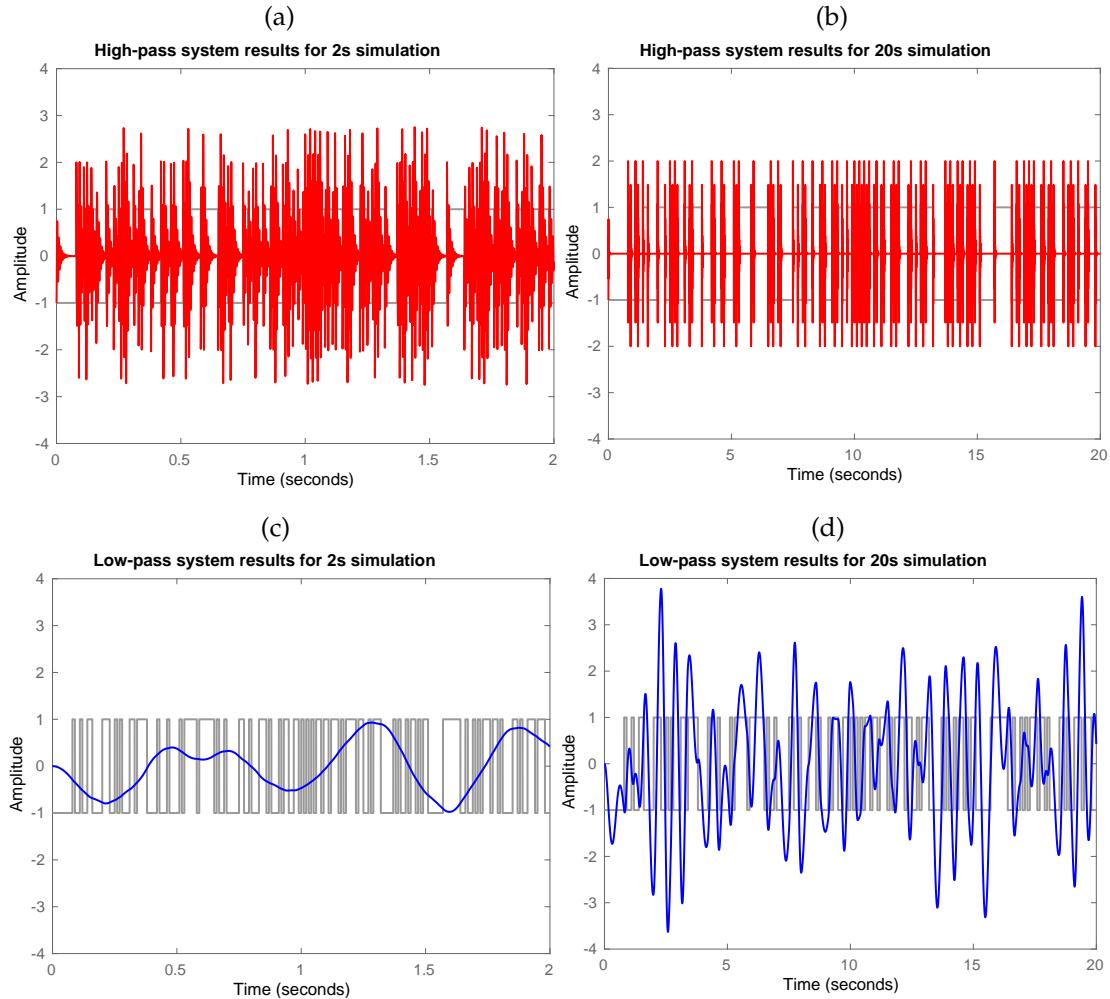


Figure 5.7: Linear simulation results of the low-pass and high-pass systems separately. Grey lines represent the PRBS input and colored lines the output. Graphs (a) and (b) with red outputs represent the results of the high-pass system; and (c) and (d) with blue outputs, the results of the low-pass system. 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.

System discretization is performed using exact discretization for inputs considered constant during each time interval to produce a discrete-time state space representation according to

$$x(t_{k+1}) = A_d(t_k)x(t_k) + B_d u(t_k) + w(t_k), \quad (5.15)$$

$$y(t_k) = C_d x(t_k) + D_d u(t_k) + v(t_k) \quad (5.16)$$

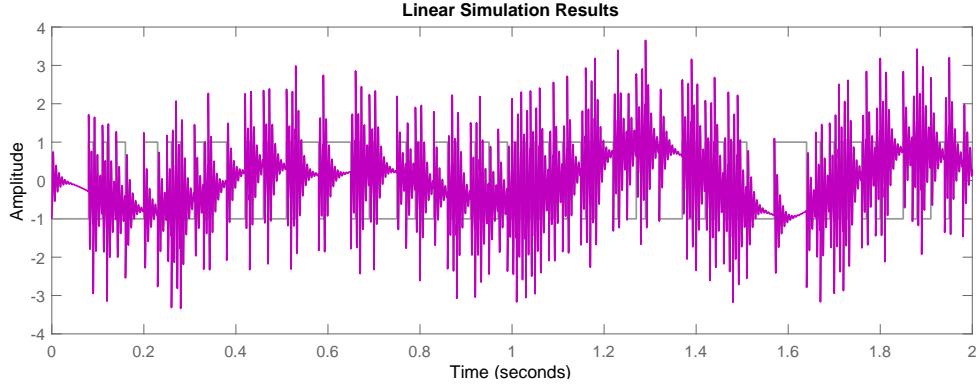


Figure 5.8: Output of the fourth order system to the PRBS input signal over a 2 seconds time frame.

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 timestamp 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 Single Realization Analysis

In this section we present one realization of the state estimation simulation for the system defined by (5.9)-(5.14), using the Kalman Filter described in Section 4.2.1 and the algorithm modifications explained in Section 4.3.

Simulation parameters are set according to $\lambda = 500$ Hz, $SNR_{\text{obs}} = 30$ dB, and $SNR_{\text{pro}} = 30$ dB, where subscript obs and pro refer to observation and process, respectively. Time step to simulate nominal system model is set as $\delta t_{\text{sim}} = 10^{-6}$ s and the regular estimation time interval is given by $T = 2 \times 10^{-3}$ s. Noise is generated as a zero mean, white, Gaussian signal with $R = 9.596 \times 10^{-4}$ and process noise variance of $Q = 9.986 \times 10^{-4} I_{4 \times 4}$. These exact values are 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 5.9.

For both systems states, we observe the degradation for the algorithm that approximates time instants t_k in the estimation process, represented by the dashed blue lines.

Figure 5.10 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 timestamp. As expected, when

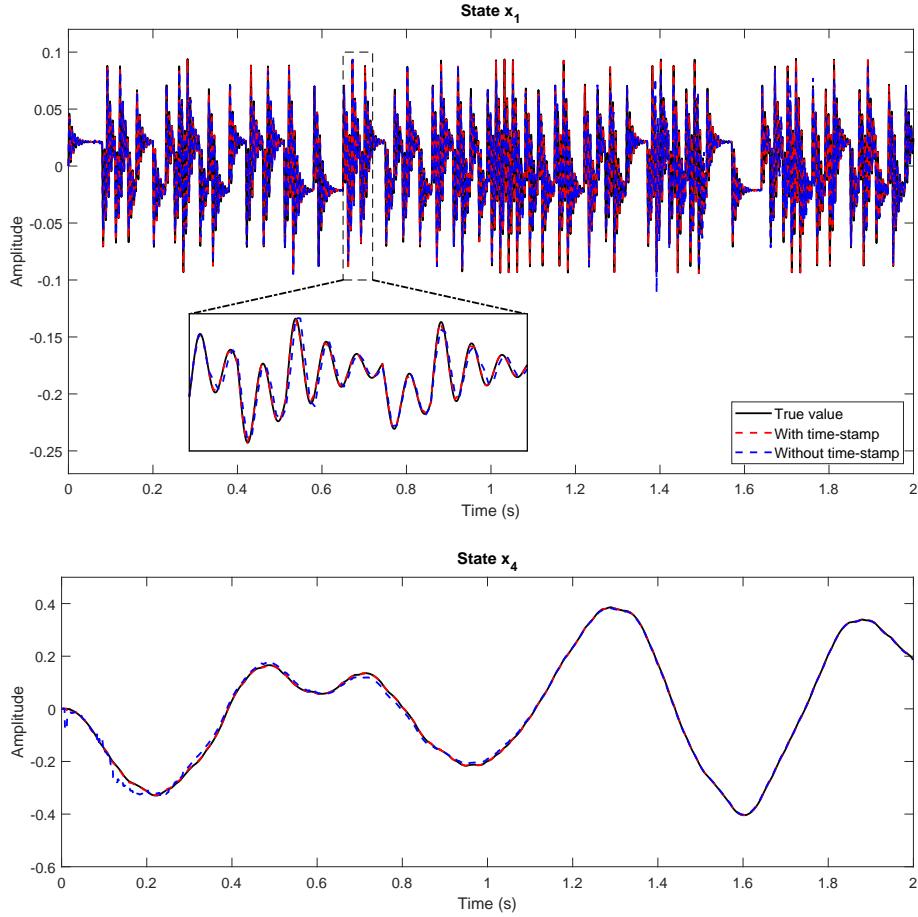


Figure 5.9: States x_1 (high-pass) and x_4 (low-pass) estimates with timestamp (---), without timestamp (- -) and true values (—).

first observation arrives, RMSE results distance themselves, in favor of the algorithm that considers timestamp. The RMSE difference increases when the next measurement (the second one) arrives. 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 5.11. 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 timestamp 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 timestamp was not available, consistency was heavily degraded, with rejection rates

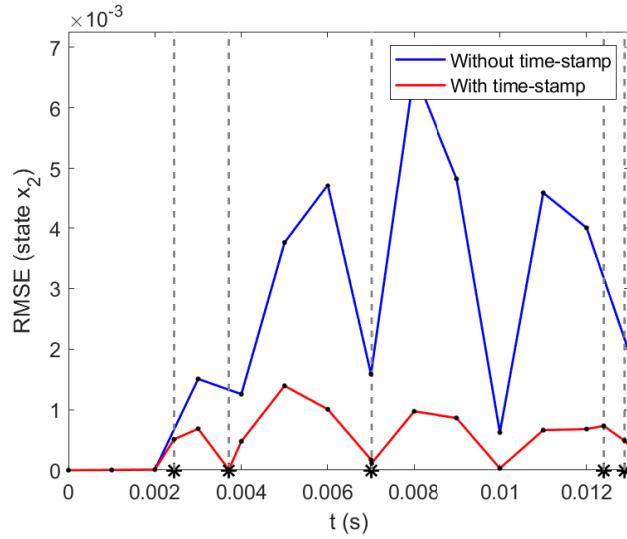


Figure 5.10: Temporal cut from 0 to 0.013 seconds, for a realization of state x_2 RMSE for both estimators, with (—) and without (—) timestamp. Vertical dashed lines match the measurement sampling instants t_k . Black dots represent the estimation instants.

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 and a 100-run Monte Carlo simulation is performed for each of the parameter set. Only the results of one state per mode are shown, namely x_1 and x_4 , for simplicity sake. We present the paired t-test results for states x_1 and x_2 in Table 5.3. Figure 5.12 encompasses all performance metrics results for all scenarios. Each column contains the performance metrics variation for each scenario.

Table 5.2: Sets of parameters for linear system simulation

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]

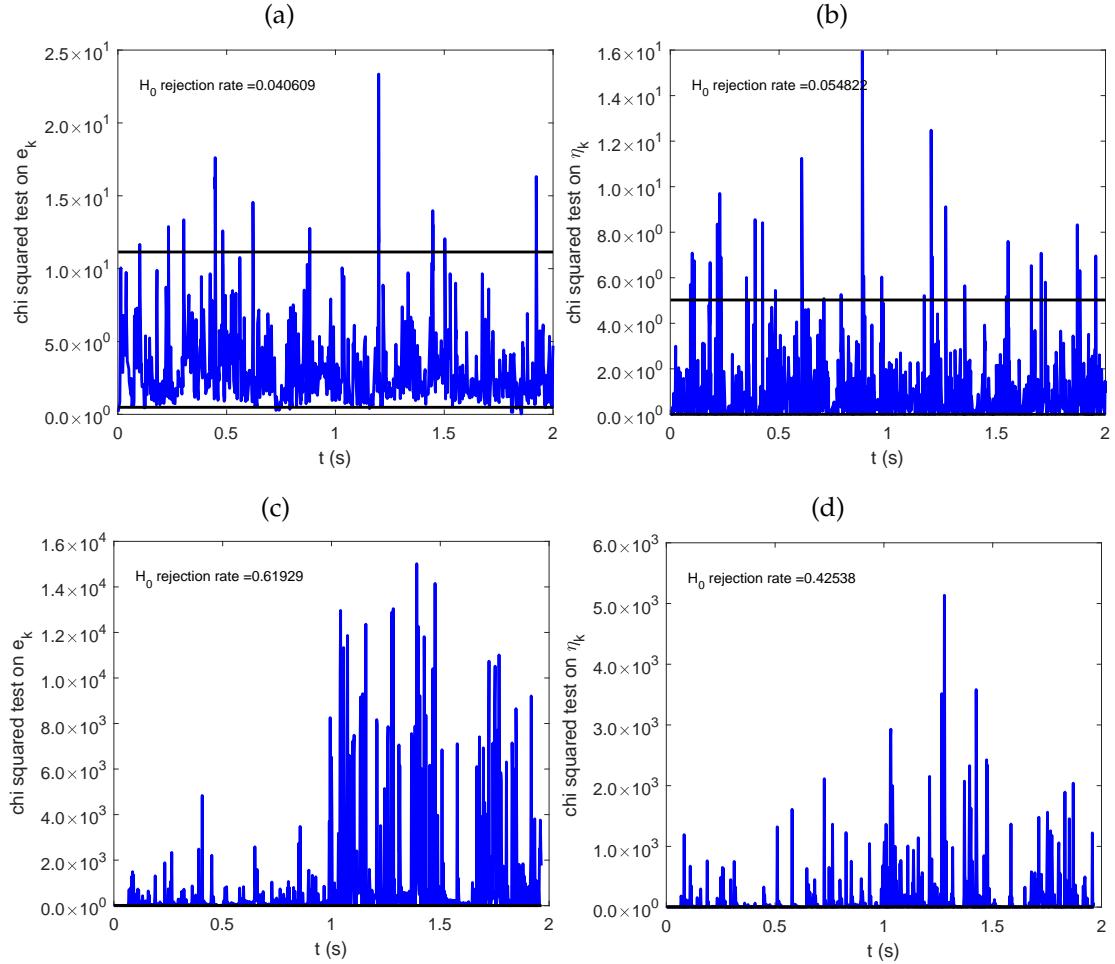


Figure 5.11: Consistency tests NEEs and NIS for the linear system state estimation with timestamps (a) and (b); and without timestamp (c) and (d). Horizontal lines define the acceptance region upper and lower limits, for a significance level $\alpha = 0.05$. In each graph, the null hypothesis H_0 rejection rate is also presented.

5.3.3 Measurement Signal-to-Noise Ratio Variation

Results from the first row of Table 5.3 suggest an increasing trend of RMSE degradation as we increase SNR levels, with an apparent upper bound limit. The mean differences of both states x_1 and x_4 have no significant variation for SNR levels as high as 40, 50 and 60 dB. On the other hand, when noise level is high, with SNR= 10 dB, the RMSE differences have much smaller practical significance, with an effect size of approximately 0.7 standard deviation for both states.

In the first column of Figure 5.12 we can observe a decreasing monotonic behavior for RMSE results of the algorithm considering timestamp, as SNR levels increase, as expected. However, the same behavior cannot be observed for the method that neglects

timestamp information. In fact, RMSE results for state x_4 reduce for higher noise levels in data. Such trend can be explained by the fact that, for higher SNR, the added error introduced by shifting time instants plays a more important role in the degradation of the estimates. When SNR decreases, its relevance is reduced, and the covariance matrix used by the algorithms are closer to the reality. Thus estimates are more consistent and for the low-pass system state x_4 results have lower RMSE.

As for NEES and NIS tests, considering timestamp produces consistent estimates for all cases, with mean rates for NEES and NIS near the 5% significance level. On the other hand, not considering timestamps produced inconsistent results in all scenarios, specially when there are lower noise levels in the data.

Finally, the shapes of the box and whiskers plots suggest a sample distribution with higher variability for accuracy results of the algorithm that neglects timestamp, which means less reliability in the estimation.

5.3.4 Average Sampling Rate Variation

Variation of performance indices for different λ values are shown in the second column of Figure 5.12 and the second row of Table 5.3. We observe the smaller differences of the means for higher λ values, suggesting that the accuracy degradation levels caused by neglecting timestamps are higher for smaller average frequencies of the aperiodic sampled observations. 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 measurement error introduced by data assimilation performed at incorrect time instants. The only exception is for state x_1 RMSE results for $\lambda = 100$ Hz. However, the sample distribution shown in Figure 5.12 (b) has such a high variability with distant outliers, that it would be hard to draw any conclusion from this scenario. The trend shown in (e), on the other hand, is very clear. However, when it comes to effect size, we do not observe a clear trend in degradation increase. Cohen's d values for state x_4 shows little difference in effect size, which is due to the variation of the standard deviations in the data from the algorithm without timestamp.

Consistency performance results in graphs (h) and (k) show that, for the simulated parameter combinations, not considering timestamp produces inconsistent estimates in all cases, while the algorithm with timestamp achieves rejection rates on NEES and NIS tests consistently closer to 5%. However, we observe an increase in the rejection rates for higher frequencies in the algorithm considering timestamp. It can be due to

numerical errors in discretization steps performed with very small time intervals. But it would require further investigation, though.

Once again, accuracy data distributions for the algorithm not considering suggest a much higher variability, impacting negatively in estimation reliability.

5.3.5 Regular and Average Irregular Time Interval Relation Variation

Results obtained from increasing α values suggest monotonic decreasing trends for the mean of the differences in RMSE for both states x_1 and x_2 , as shown in the third line of Table 5.3 and in Figure 5.12 (c) and (f). 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 less important it is to assimilate data at the correct time instants t_k . In fact, the effect size of state x_1 for $\alpha = 5$ shows a very small practical significance in the mean difference from both algorithms. Similarly to the scenario where we vary λ values, it can be explained by the reduction in approximation errors of $\hat{y}_i \approx y(t_k)$ and to the fact that we considered the same SNR level for both input and output in the simulation.

By comparing the accuracy values of states x_1 and x_4 , shown in Table 5.3, we can infer a more relevant decrease in accuracy degradation for the state of the high-pass system. Whereas its mean difference decreased almost 7 times by increasing α from 1 to 5, that is the mean of the differences went from 6.85×10^{-4} to 0.98×10^{-4} , the mean values of state x_4 decreased approximately 3.5 times, from 16×10^{-3} to 4.35×10^{-3} . Same behavior is observed in effect size values, with a reduction of almost 4 times for state x_1 compared to a decrease in approximately 2 times for x_4 .

Again, consistency test results show underestimated errors for the algorithm without timestamp systematically, whereas considering timestamp produces much more consistent estimates. An exception occurred for NEES test results, when α is as high as 5. We can infer that it is due to a more frequent estimation using only the forecast step, since observation time intervals are more sparse in comparison to the regular estimation time interval.

Table 5.3: Linear system paired t-test results and effect size estimates for RMSE of states x_1 and x_4 . 95% confidence intervals for the mean of the differences μ_D and for Cohen's d are shown. p-values for t-test results were all lower than 10^{-9} , except for state x_1 considering $\alpha = 5$, for which it was 3.6×10^{-4} .

		Scenarios		State 1 (RMSE difference)		State 4 (RMSE difference)	
				μ_D	Cohen's d	μ_D	Cohen's d
SNR (dB)	10			$[2.7, 5.0] \times 10^{-4}$	$[0.39, 0.97]$	$[2.3, 3.8] \times 10^{-3}$	$[0.52, 1.1]$
	20			$[5.0, 6.9] \times 10^{-4}$	$[0.97, 1.6]$	$[11, 14] \times 10^{-3}$	$[1.6, 2.3]$
	40			$[7.4, 9.1] \times 10^{-4}$	$[1.6, 2.3]$	$[18, 21] \times 10^{-3}$	$[2.3, 3.1]$
	50			$[8.8, 10] \times 10^{-4}$	$[1.9, 2.6]$	$[18, 21] \times 10^{-3}$	$[2.8, 3.7]$
	60			$[8.3, 10] \times 10^{-4}$	$[1.6, 2.3]$	$[18, 21] \times 10^{-3}$	$[2.6, 3.4]$
λ (kHz)	0.1			$[4.6, 7.3] \times 10^{-4}$	$[0.58, 1.2]$	$[3.7, 4.4] \times 10^{-2}$	$[2.0, 2.7]$
	0.3			$[6.5, 8.2] \times 10^{-4}$	$[1.4, 2.0]$	$[3.1, 3.7] \times 10^{-2}$	$[2.1, 2.9]$
	0.5			$[6.1, 8.1] \times 10^{-4}$	$[1.1, 1.7]$	$[1.6, 1.9] \times 10^{-2}$	$[2.3, 3.1]$
	1			$[2.1, 3.5] \times 10^{-4}$	$[0.46, 1.0]$	$[0.66, 0.78] \times 10^{-2}$	$[2.0, 2.7]$
α	1			$[5.9, 7.8] \times 10^{-4}$	$[1.1, 1.8]$	$[15, 17] \times 10^{-3}$	$[2.5, 3.2]$
	2			$[3.1, 4.2] \times 10^{-4}$	$[0.96, 1.6]$	$[8.9, 11] \times 10^{-3}$	$[1.9, 2.6]$
	3			$[1.6, 2.5] \times 10^{-4}$	$[0.56, 1.1]$	$[5.6, 6.7] \times 10^{-3}$	$[1.7, 2.4]$
	5			$[0.46, 1.5] \times 10^{-4}$	$[0.088, 0.65]$	$[3.8, 4.9] \times 10^{-3}$	$[1.3, 2.0]$

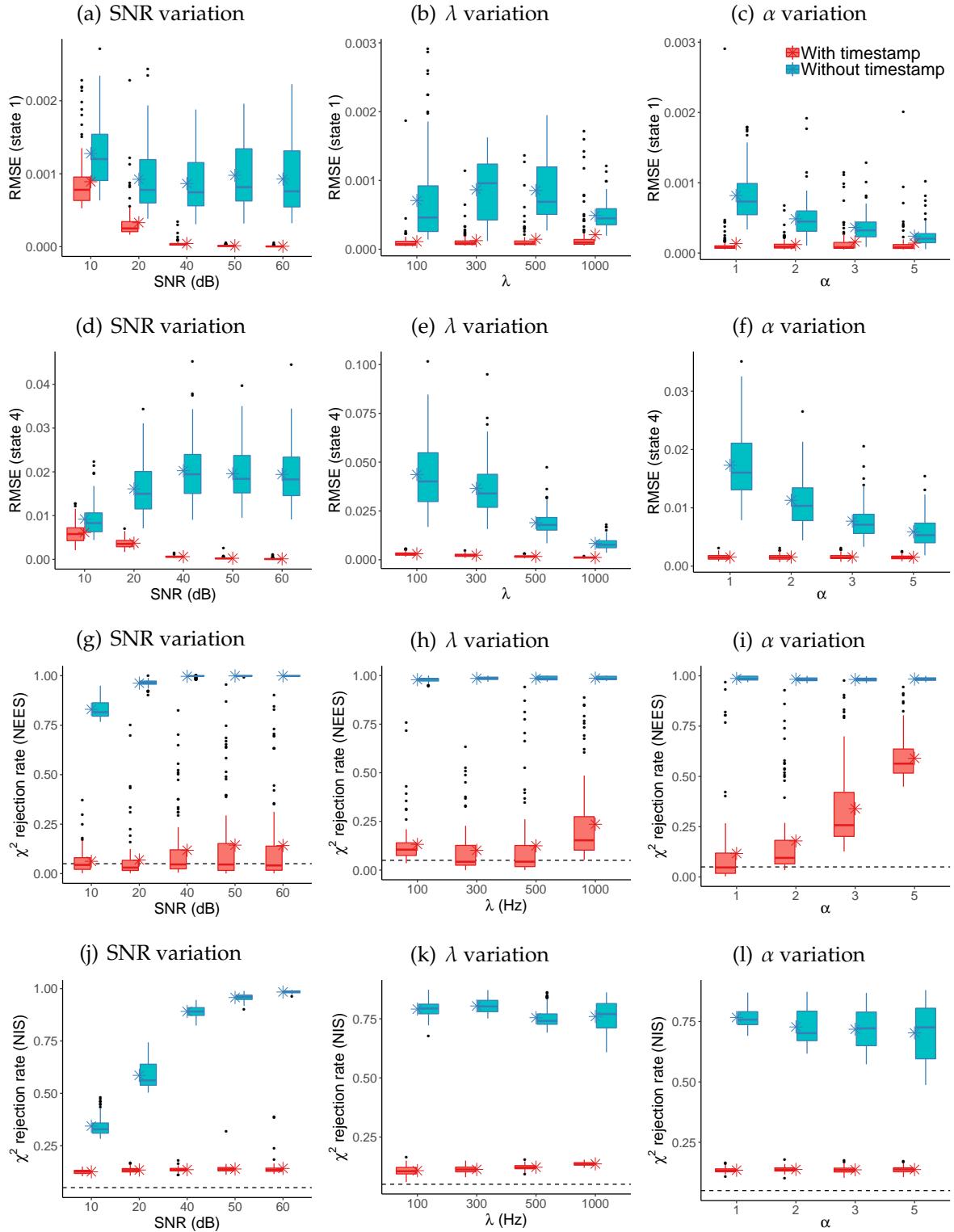


Figure 5.12: Box and whiskers plots for the linear system estimation results, as a function of SNR (a,d,g,j), λ (b,e,h,k) and α (c,f,i,l), considering (red) and not considering (blue) timestamps. Mean values are presented in asterisks. Horizontal gray dashed lines for consistency results represents the 5% significance level limits.

5.4 Nonlinear System: Unicycle Position

We now consider a nonlinear system given by a nonholonomic 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. On the other hand, inertial sensors responsible for input data, such as gyroscopes and accelerometers transmit data regularly, but usually with lower SNR values.

5.4.1 System Description

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

$$\begin{aligned}\dot{p}_x &= v \cos(\theta), \\ \dot{p}_y &= v \sin(\theta), \\ \dot{\theta} &= u_1(t), \\ \dot{v} &= u_2(t),\end{aligned}\tag{5.17}$$

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 5.13 shows a schematic of the robot and its states,

The system described by (5.17) is discretized by a fourth order Runge-Kutta method as described in Section 4.1 and the discrete-time 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 $y(t_k) \in \mathbb{R}^2$

$$y(t_k) = \begin{bmatrix} p_x(t_k) \\ p_y(t_k) \end{bmatrix} + v(t_k),\tag{5.18}$$

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 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 .

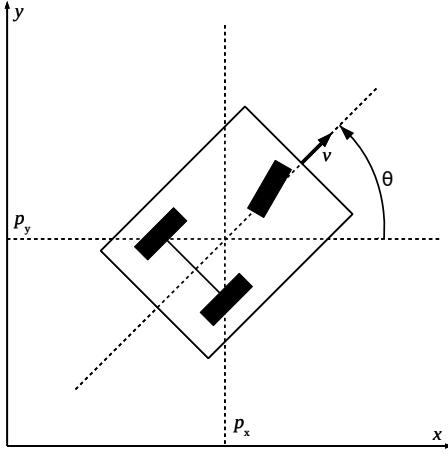


Figure 5.13: Nonholonomic robot system representation. The system states p_x , p_y , v and θ are highlighted.

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

$$u_i = \tilde{u}_i - w_i, \quad (5.19)$$

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 $\delta t_{\text{sim}} = 10^{-4}$, generating 600.000 samples to mimic the continuous system dynamics. Input signals are generated arbitrarily according to Figure 5.14.

Accuracy metrics for the unicycle position system is a modification of the RMSE defined in (4.112), to reflect error in xy -plane position, given by:

$$RMSE_{\text{position}} = \frac{\sum_{i=1}^N \sqrt{(\hat{p}_{x,i} - p_{x,i})^2 + (\hat{p}_{y,i} - p_{y,i})^2}}{N}, \quad (5.20)$$

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 Single Realization Analysis

Figure 5.15 (a) 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

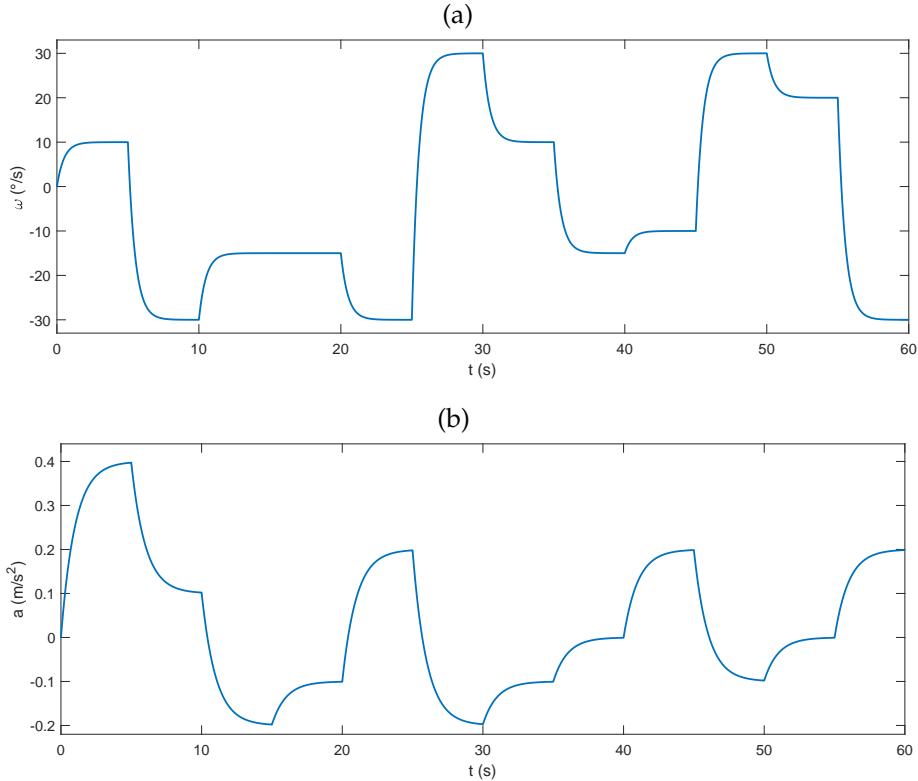


Figure 5.14: Simulation input signals. (a) shows temporal sequence of angular velocity ω and (b) shows linear acceleration a .

as red dots, with signal-to-noise ratio of $\text{SNR}_y = 30$ dB and $\lambda = 3.33$ Hz. The dashed blue line represents UKF estimation, considering timestamp, $\alpha = 5$ and $\text{SNR}_u = 10$ dB.

Figure 5.15 (b) shows a timespan from 0 to 1.5 seconds of a position RMSE realization, considering $\lambda = 0.5$, $\alpha = 2$, $\text{SNR}_y = 40$ dB and $\text{SNR}_u = 20$ dB, for the UKF considering and not considering timestamp. 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. The third measurement, 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 the instant of the fourth measurement, around 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 parameter combination, for the scenarios described in Section 5.2. Simulation parameters

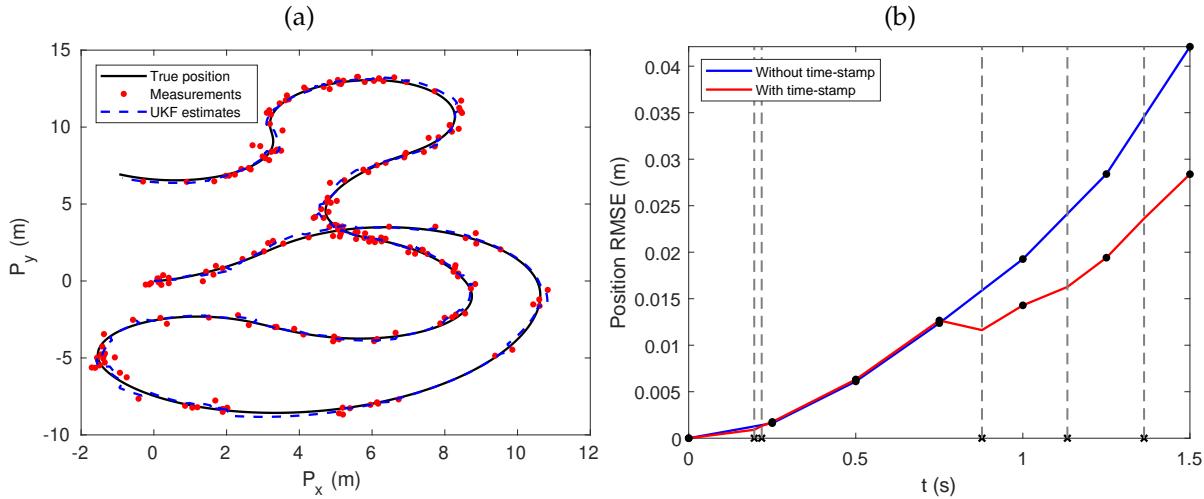


Figure 5.15: (a) True position (—), noisy measurements (•) and UKF estimates (---) considering timestamp, for a measurement noise of $\text{SNR}_y = 30 \text{ dB}$, $\lambda = 0.3 \text{ s}$ and $\alpha = 5$. (b) Temporal cut from 0 to 1.5 seconds, for a realization of J of both estimators, with and without timestamp. 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.

used for the nonlinear system are shown in Table 5.4. For all cases, input SNR is constant at 20 dB. Paired t-test results and effect size of the mean differences for position RMSE are shown in Table 5.5. Figure 5.16 summarizes information of all results, with box and whiskers plots of numerical simulation data for each scenario. First row of graphs shows accuracy results for position RMSE, as defined in (5.20). Second and third rows present the consistency results for NEES and NIS tests, respectively, as described in Section 4.4. Red plots are for the algorithm that considers timestamp and blue for the one without timestamp.

Table 5.4: Sets of parameters for unicycle position estimation simulation parameters

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]

5.4.3 Measurement Signal-to-Noise Ratio Variation

Results from the first row of Table 5.5 and the first column of Figure 5.16 show the variation in performance for the different measurement SNR levels simulated. Both algorithms, in most cases, have their estimate error increased, when measurement data is contaminated with higher noise levels, as expected. We observe from Table 5.5, however, that the mean differences are higher when there is less noise, in favor of considering timestamp in the estimation process. The effect size results also suggest an upper bound for such difference. It can be explained similarly to what was observed in the linear system. The higher the SNR levels, 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. In fact, there is smaller practical significance in the results for low SNR levels, as shown by Cohen's d values.

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 might be due to numerical errors and approximations caused by very small measurement covariances, when there is virtually no noise in data.

From the box and whiskers plots for position RMSE, we can observe once again higher variability in the results obtained from neglecting timestamps. A more significant presence of outliers is also apparent. Both aspects indicate less reliable estimation performance.

5.4.4 Average Sampling Rate Variation

As for different λ values, variations of the metrics are presented in the second column of Figure 5.16 and the mean values of the differences in position RMSE are shown in the second row of Table 5.5. Results suggest a slightly higher performance degradation for more sparse observation time intervals, with a 3 to 4 cm variation in difference of position RMSE. With an exception of the scenario with $\lambda = 10$ Hz, the trend is monotonic. However, due to a much higher variability in result distributions for both accuracy and consistency indices, it is hard to draw meaningful conclusions and further investigation is required.

NEES and NIS test results suggest that inconsistency levels for the algorithm that neglects timestamp decreases as average observation frequency increases, whereas considering timestamp produced consistent estimates most of the time.

Another interesting observation is the amount of data outliers, apart from higher data variability, for the results obtained by neglecting timestamp in the estimation process. That can be very detrimental when performance reliability is needed.

5.4.5 Regular and Average Irregular Time Interval Relation Variation

Finally, results from the last scenario, that is variation of α , are shown in the third column of Figure 5.16 and third row of Table 5.5. Mean difference values suggest estimation performance degradation is more significant for higher α values. Effect size results do not support such conclusion, due to an increase in data variability, keeping variation almost constant.

However, for considering the mean differences of the accuracy metric, we observe an opposite trend for the nonlinear system in comparison to the linear system. Whereas mean values of the differences decreased for higher α values in the linear case, here they increase. 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, then the data assimilation steps are relatively more frequent. That might be one explanation for the obtained results. However, we need a deeper analysis to draw more meaningful conclusions.

We also observe that the results from the algorithm with timestamp remained approximately constant with very little variability throughout all cases. It indicates that when exact time instants t_k are available to the estimator, the ratio given by α has small impact on performance.

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

Table 5.5: Nonlinear system paired t-test mean difference results in cm and effect size estimates for position RMSE. 95% confidence intervals for the mean of the differences μ_D and for Cohen's d are shown. p-values for t-test results were all lower than 10^{-100} due to the huge amount of samples, given by the 1000-run Monte Carlo simulation.

	Scenarios	Position (RMSE difference)	
		μ_D (cm)	Cohen's d
SNR (dB)	10	[3.0, 3.4]	[0.76, 0.94]
	20	[4.2, 4.8]	[0.83, 1.0]
	40	[7.8, 8.7]	[1.0, 1.2]
	60	[9.8, 11]	[1.2, 1.4]
	80	[11, 13]	[1.2, 1.4]
	100	[16, 18]	[1.2, 1.3]
λ (kHz)	1.67	[10, 12]	[1.1, 1.3]
	2	[9.1, 10]	[1.0, 1.2]
	2.5	[8.4, 9.7]	[0.82, 1.0]
	3.33	[6.9, 7.9]	[0.79, 0.97]
	5	[6.0, 6.9]	[0.78, 0.96]
	10	[7.7, 8.7]	[1.0, 1.2]
α	1	[6.2, 6.8]	[1.2, 1.4]
	2	[7.4, 8.3]	[1.0, 1.2]
	5	[17, 18]	[1.3, 1.5]
	10	[33, 35]	[1.9, 2.1]

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 of each parameter: (i) SNR; (ii) λ ; and (iii) α .

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

As expected, the average performance metrics for the algorithm considering timestamp are always higher in comparison to the metrics from the algorithm without timestamp, considering a confidence level of 95%. Accuracy performance for SNR variation suggests that for lower noise levels or higher SNR, the relevancy of considering or neglecting timestamp 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 has small practical significance.

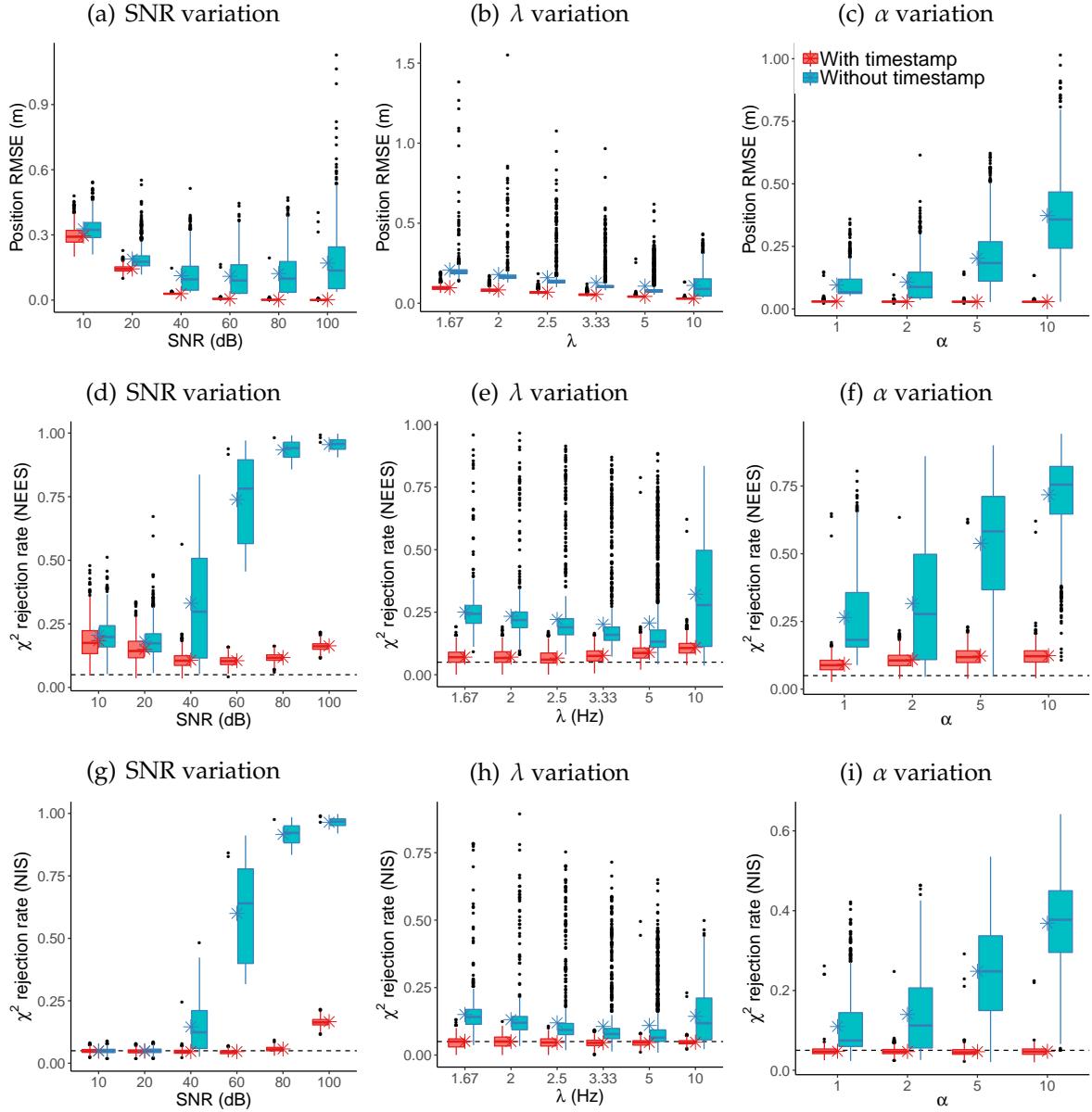


Figure 5.16: Box and whiskers plots for the nonlinear system estimation results, as a function of SNR (a,d,g), λ (b,e,h) and α (c,f,i), considering (red) and not considering (blue) timestamps. Mean values are presented in asterisks. Horizontal gray dashed lines for consistency tests represents the 5% significance level limits.

For different λ values, the algorithm that considers timestamp maintained its performance metrics almost constant, specially for consistency tests, as it was expected. Results from the algorithm without timestamp indicate that more sparse observation time intervals deteriorate estimation performance, due to higher approximation errors. When it comes to effect sizes, however, such behavior is not so clear.

Finally, performance values for the variation of α values show different behaviors

for the linear and nonlinear system for the algorithm without timestamp. The linear case indicates an increase in state estimates RMSE, for smaller α values, provided that both input and measurement data have the same SNR level. 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, mean differences of position RMSE show an opposite trend, with effect size results almost constant through the different simulated scenarios. One explanation might be the fact that we consider different SNR levels in measurement and input data, that is $SNR_{\text{observation}} = 40$ dB and $SNR_{\text{input}} = 20$. Therefore, although the added measurement 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

6.1 Project Overview

In this study, we investigate how neglecting timestamps of irregularly sampled measurements may affect state estimation performance, under different conditions.

The research motivations are discussed in Chapters 2 and 3, where sensor fusion methods and irregular sampling schemes are reviewed, respectively. We note how sensor fusion, as a field of science, has been experiencing significant growth due to cheaper sensors and communication devices. Bigger and more complex sensor networks are more and more common, and so are the related challenges, such as irregular sampling. Then we define the irregular sampling problem and review the main schemes studied in the literature. Timestamp information is often needed in order to take these irregularities into account in data fusion processes. We note that sometimes due to computational complexity or the need of investment in time synchronization techniques, such assumption is not always guaranteed. Thus we conclude the literature review focusing in the study of sensor fusion applied to state estimation and how it is affected when timestamp is neglected.

In Chapter 4 we describe the algorithms used in the numerical simulations. We begin by describing the discrete-time representation of sampled-data systems, considering variable time steps in the discretization process, since measurements are sampled aperiodically. We also review the most popular state estimation algorithm, that is the Kalman Filter (KF), and one of its variation for nonlinear cases, the unscented Kalman Filter (UKF). We describe them as a data fusion process, using the Bayesian framework. Then we describe the sampling irregularity model, which is considered to be given by a Poisson process, meaning that the waiting time between two consecutive measurements is a exponential random variable (RV). Moreover, we describe the adaptations made to the algorithms due to the aperiodic sampled measurements. Whenever timestamps are not available, the measurements are considered to be taken in the next regular estimation interval. A first glimpse of the added error introduced by neglecting

timestamps is also presented in this chapter. We end it by defining performance metrics used in the simulation, that cover both accuracy and consistency.

Numerical results are presented in Chapter 5. We begin by studying the error effects on measurements introduced by shifting the time instants they were taken, and how these errors are related to signal-to-noise ratios (SNR) and system dynamics. Based on that, we propose a simulation setup, where system parameters are varied to assess their relation to state estimation performance, knowingly: (i) measurement noise levels, or SNR; (ii) average sampling rate of measurements, referred to as λ ; and (iii) the relation between λ and the regular sampling rate of the estimation, referred to as α . Two systems are chosen as case studies: a fourth-order linear time invariant system; and a nonlinear system given by a nonholonomic moving unicycle robot, whose position in the xy -plane must be estimated. We illustrate the state estimation problem under irregular sampling by one realization of the estimates. To assess the performance differences from the algorithms that consider and neglect measurement timestamps, we run multiple simulations and calculate the performance metrics mean values and effect sizes in a confidence interval of 95%, which are then compared and discussed.

6.2 Main Results and Contributions

We consider the research objectives introduced in the beginning of this work to be achieved. As for the first one, an extensive review in sensor fusion as a field of science is provided, considering its motivations and the taxonomies proposed in literature. We also categorize and discuss most of the irregular sampling models studied by state estimation applications.

We then choose Kalman filter and the unscented Kalman filter as state estimation methods, presenting their implementation details. By modeling the measurements random time instants as a Poisson process, we cover a very generic irregular sampling problem. The motivations behind it lie on large sensor networks applications, where data is transmitted periodically but unsynchronized. Resulting arrival times can be approximated by an exponential random variable. We go through the adaptations to the state estimation algorithms to the situations when timestamp is part of the data packet, and when they are not available. Therefore, second objective is also covered.

The analysis on the error that is added to the measurement data, by shifting the irregular time instants, substantiated the design of the simulation setup, which is the third objective of this study. We end up with a framework that suggests that the

degradation in state estimation performance by neglecting timestamp depends on SNR levels, on the average sampling rate of measurements λ , and the relation between the average sampling rate of measurements with the regular sampling rate of estimation α .

By applying the proposed framework to two different systems, one being linear and the other nonlinear, we test if the effects can indeed be assessed, fulfilling the last objective. For the case studies considered we observe some significant performance variation for some parameter sets, whereas for other combination of system parameters, neglecting timestamp did not play an important role. When system SNRs increase to a certain level, considering timestamp or not becomes irrelevant. Results suggest that timestamp information might also be neglected for high sampling rates. However, what is considered to be high will depend on the system dynamics. Another conclusion we draw from results is that performance degradation for the variation of the relation between measurement and input sampling rates α might depend on the relation between their SNR levels. As a final note, we also observe a higher variability in performance metrics mean values for the algorithm that does not consider timestamp, which can be a problem if consistency is needed.

We must also be self-critical and question what could have been more consistent or realistic in our study. First, the irregular sampling model, although generic, could have been closer to reality by including time delay on data transmission. Such configuration would better justify the lack of timestamp in the data packet. However, time delay in state estimation without timestamps seems to have a similar effect to the problem we formulated, which is a time shift in measurement time instants. Adaptations to the algorithms, on the other hand, are much more complex. Additionally, as we observe in the results for α variation, different combination of SNR in process and observation models could have been carried out to evaluate a joint contribution. And finally, since we argue that neglecting timestamp will introduce an error to the measurements, then we could have tested state estimation algorithms with an artificially higher covariance of the observation model. We could also have tried some error compensation method that is related to an estimate of the output derivative, since they are related.

6.3 Future Work

From the improvement points discussed on the last section and some insights provided by the results, we identified some opportunities for further investigation, which

are summarized as follows:

1. A very clear relation between the error and the derivative of the signal was identified, one being directly proportional to the opposite of the other. We suggest the investigation of an error compensation algorithm to be used in state estimation;
2. After using the proposed framework in simulations, a tuning routine seems promising, increasing the trace of the covariance matrix of the measurement noise. Although we showed that the added error is not white Gaussian, better state estimation performance might be achieved;
3. In this study we did not include time delay as part of the irregular sampling problem. In real applications where transmission delays are present, new aspects of the systems might impact the degradation in performance and we suggest further investigation;
4. We considered only Kalman filter and its unscented variation as state estimation algorithms in this study. We suggest the use of particle filter based methods to investigate if it will provide similar results. We argue that, since they are more promising for non-Gaussian noise, they could be more robust for scenarios where the signal-dependent error, introduced by neglecting timestamp, is more relevant;
5. Further investigation in the variation of α for different combinations of SNR in process and observation models might also be interesting.

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 Skövde.
- 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.
- 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 H₂-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.
- Menegaz, H. M. T., Ishihara, J. Y., Borges, G. A., and Vargas, A. N. (2015). A Systematization of the Unscented Kalman Filter Theory. *IEEE Transactions on Automatic Control*, 60(10):2583–2589.
- 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.
- 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.
- 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.
- Wilk, M. B. and Gnanadesikan, R. (1968). Probability plotting methods for the analysis of data . *Biometrika*, 55(1):1–17.
- 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.
- Yen, J. (1990). Generalizing the Dempster-Shafer Theory to Fuzzy Sets. *IEEE Transactions on Systems, Man and Cybernetics*, 20(3):559–570.
- 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.