



UNIVERSIDAD DE GUANAJUATO

CAMPUS IRAPUATO SALAMANCA

DIVISIÓN DE INGENIERÍAS

**Design of unbiased state estimators for
WSNs with consensus on measurements
and estimates and improved robustness**

TESIS DOCTORAL

QUE PARA OBTENER EL GRADO:

DOCTOR EN INGENIERÍA ELÉCTRICA

PRESENTA:

M.C. MIGUEL ANGEL VÁZQUEZ OLGUÍN

ASESORES:

Dr. YURIY S. SHMALIY

DR. OSCAR G. IBARRA MANZANO

Salamanca, Gto., a 22 de Octubre del 2019.

M. en I. HERIBERTO GUTIÉRREZ MARTÍN
JEFE DE LA UNIDAD DE ADMINISTRACIÓN ESCOLAR
PRESENTE.-

Por medio de la presente, se otorga autorización para proceder a los trámites de impresión, empastado de tesis y titulación al alumno(a) M. en C. Miguel Ángel Vázquez Olguín del Programa de Doctorado en Ingeniería Eléctrica y cuyo número de NIA es: 145583 del cual soy director. El título de la tesis es: Design of unbiased state estimators for WSNs with consensus on measurements and estimates and improved robustness

Hago constar que he revisado dicho trabajo y he tenido comunicación con los sinodales asignados para la revisión de la tesis, por lo que no hay impedimento alguno para fijar la fecha de examen de titulación.

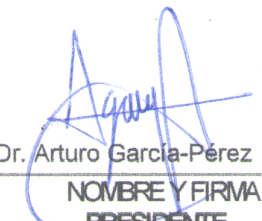
ATENTAMENTE


Prof. Yuriy S Shmaliy

NOMBRE Y FIRMA
DIRECTOR DE TESIS
SECRETARIO


Dr. Oscar G Ibarra-Manzano


NOMBRE Y FIRMA
DIRECTOR DE TESIS


Dr. Arturo García-Pérez

NOMBRE Y FIRMA
PRESIDENTE


Dra. María del Rosario Baltazar Flores

NOMBRE Y FIRMA
VOCAL


Dr. Miguel Octavio Arias-Estrada

NOMBRE Y FIRMA
VOCAL


Dr. J. Amparo Andrade-Lucio

NOMBRE Y FIRMA
VOCAL

Abstract

Wireless sensor networks (WSNs) is a technology with important developments in recent years. Its incursion in areas such as healthcare, industry and services has been steadily increasing, mainly due to the miniaturization of electronics and the growing acceptance of cyber-physical systems. However, a very important subject of research continues to be the development of estimators with the robustness needed for the harsh conditions associated with the WSNs applications. Moreover, such estimators should comply with the unique characteristics imposed by the WSNs like scalability, energy saving and redundancy, while maintaining a consensus on the network. A very popular algorithm for optimal estimation is the Kalman filter (KF). Many works have implemented it as a sensor fusion technique in WSNs, due to its optimality. However it has been proven that it can not guarantee the robustness needed in real life implementations. In this work we developed a set of robust estimators based on unbiased finite impulse response (UFIR) filters to address the lack of robustness of the popular KF. The developed filters are adequate to be implemented in WSNs. The algorithms have been tested against similar filters based on KF with simulated and real data, showing better results in terms estimation error reduction, where the smallest improvement was of 1.4 percent in terms of the root mean squared error (RMSE). We even produce accurate results in applications where KF could not be implemented. The developed filters attained better robustness against miss-model errors, unknown statistics and missing measurements.

Acknowledgments.

To god, as it is my strongest belief that nothing happens without His will.

To my wife Norma and my daughters Valentina and Verónica, for their unconditional love and support.

To my parents Angel and Ma. de la Luz and my brother and sister, David and Luz Alejandra. Their support and unconditional love made this achievement possible.

To my advisors, Dr. Yuriy S. Shmaliy and Dr. Oscar G. Ibarra, I express my most sincere gratitude for their guidance and patience. To Dr. Yuri S. Shmaliy for being the principal architect of my doctoral formation through his vast knowledge and patience. To Dr. Oscar G. Ibarra, for his support and advise though difficult times.

To my thesis committee Dr. Arturo García Pérez, Dr. José Amparo Andrade Lucio, Dra. María del Rosario Baltazar Flores, Dr. Miguel Octavio Arias Estrada.

To all the remarkable people that I met during these four years. Dra. Dora Luz Almanza and Dr. Mario Ibarra, my colleges Carlos Lastre, Jose Luis Contreras, Guillermina Vivar, Elizabeth Martinez, Alberto Lopez, Jorge Minjares, Sandra Marquez, Karen Uribe, Lucy Tetuán, Minerva Yopez, Pedro García, Alejandra Cruz and all those who granted me their friendship.



To the University of Guanajuato and to the Engineering Division of the Campus

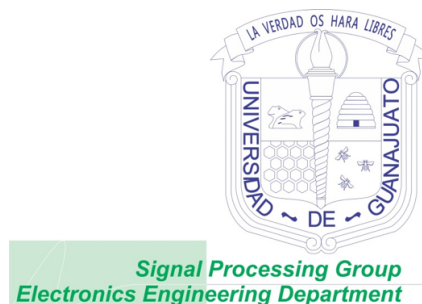
Irapuato-Salamanca, for the academic formation received during the Doctor of Electrical Engineering program.



To the National Council of Science and Technology (Consejo Nacional de Ciencia y Tecnología) CONACYT for the financial support provided during the four years of the doctoral program through the scholarship No.426428/CVU 226854.



To the Institute of Technology of Leon, and to all the professors that are part of the electronics-mechatronics academy for their support during my doctoral formation.



To the Signal Processing Group of the Electronics Engineering Department for the resources, support, fellowship and parties carried out during the doctoral program.

Contents

Title Page	i
Abstract	ii
Acknowledgments	iii
Contents	viii
List of Figures	xii
List of Tables	xiii
Acronyms	xv
1 Introduction	1
1.1 Background	1
1.1.1 Optimum and Robust Estimators	1
1.1.2 Multisensor Fusion	6
1.2 Hypothesis	9
1.3 Motivation	9
1.4 Objectives	11
1.5 Scope	11
2 Foundations	13
2.1 Smart Sensors	13
2.1.1 Power unit	13
2.1.2 Sensing unit	14
2.1.3 Processing unit	14
2.1.4 Communication unit	15
2.2 Wireless Sensor Network	15

2.2.1	Definition	15
2.2.2	Topologies	17
2.3	Distributed consensus estimation on WSNs	20
2.3.1	Algebraic graph theory	21
2.3.2	Average consensus protocol	21
3	Design of UFIR Filter for Smart Sensors	23
3.1	Introduction	23
3.2	Measured Quantity and Problem Formulation	26
3.3	UFIR Filtering Algorithm	28
3.3.1	Predictive Iterative UFIR Filtering Algorithm	29
3.4	Robustness of UFIR Filter and KF	31
3.4.1	Imprecisely Defined Noise Statistics	31
3.4.2	Temporary Model Errors	33
3.4.3	Temporary Measurement Errors	33
3.5	Experimental Verification	34
3.5.1	CO Concentration	34
3.5.2	Temperature Estimation with Missing Data	39
4	Design of UFIR Filter with Consensus on Measurements for WSNs	43
4.1	Introduction	43
4.2	Model and Problem Formulation	45
4.3	Distributed UFIR Filtering	47
4.3.1	Centralized UFIR Filter	48
4.3.2	Micro-UFIR Filter	49
4.4	Robustness of micro-Filters	52
4.4.1	Imprecisely Defined Noise Statistics	52
4.4.2	Temporary Model Errors	53
4.4.3	Temporary Measurement Errors	53
4.5	Example of Applications	54
4.5.1	Tracking Under Incomplete Information About Noise	56

4.5.2	Effect of Unspecified Impulsive Noise	59
5	Design of UFIR Filter With Consensus on Estimates for WSNs	61
5.1	Introduction	61
5.2	Model and Problem Formulation	63
5.3	dUFIR Filter with Consensus on Estimates	65
5.3.1	Batch dUFIR Filter	66
5.3.2	Optimum Consensus Factor λ_k^{opt}	67
5.3.3	Iterative dUFIR Algorithm Using Recursions	68
5.4	A Comparison of dUFIR and dKF Algorithms	69
5.5	Examples of Applications	72
5.5.1	Maneuvering Object Tracking	72
5.5.2	UVG Localization over WSN for Measured Ground Truth	74
6	Design of UFIR Filter With Consensus on Estimates for unstable WSNs	83
6.1	Introduction	83
6.2	Moving Object Model in Distributed WSNs and Problem Formulation	85
6.3	Tracking Filtering Algorithms with Consensus on Estimates	86
6.3.1	Distributed KF Algorithm	87
6.3.2	Distributed UFIR Filter Algorithm	87
6.4	Maneuvering Object Tracking with Missing Data	94
6.5	Vehicle Localization over WSN with Missing Data and Time-Varying Model Noise	96
6.6	Environmental Temperature Monitoring	104
7	Comparison with KF, H_∞ and Generalizations	110
7.1	Conclusions	110
7.2	Discussion	114
7.3	Future Work	115
	Bibliography	116
	Appendix A Derivation of the Consensus Factor λ_k^{opt}	131

Appendix B Derivation of the Recursion for α_l	133
Appendix C Derivation of the Recursion for β_k	134

List of Figures

1.1	Illustration of the estimation process.	1
1.2	Kalman filter iterative process.	3
1.3	Effects of a badly tuned KF.	4
1.4	Effects of a badly tuned H_∞	5
1.5	Better robustness of UFIR filter.	6
1.6	Consensus on measurements from [1]	7
1.7	Consensus on estimates from [1]	8
1.8	Consensus on Information from [2]	8
1.9	H_∞ consensus results from [3]	9
2.1	Basic representation of a smart sensor.	14
2.2	WSN presenting a star topology.	18
2.3	WSN presenting a tree topology.	19
2.4	WSN presenting a mesh topology.	20
3.1	Main body of a smart sensor that is responsible for sensing, signal conditioning, ADC, and signal processing. The complete architecture may comprise a DAC, communication units, control units, and an ID tag.	24
3.2	Two sources of increased carbon monoxide concentration in air: industrial pollution and car pollution.	35
3.3	20 days reference (Ref) and real (Sensor) measurements of carbon monoxide concentration in part of mg/m^3 [4, 5].	36

3.4	CO concentration measured by a sensor (Sensor) and estimated using Algorithm 1 (UFIR filter) on a horizon of $N = 12$ points. Data at each point are one-hour averaged.	36
3.5	Spectral features of CO concentration estimated using Algorithm 1 with $N = 12$ and $N = 24$: (a) a_{0k} , (b) a_{1k} , and (c) a_{2k}	37
3.6	CO concentration measured by a sensor (Sensor) and estimated using the prediction feature of Algorithm 1. The horizon (a) $N = 12$ and (b) $N = 24$ are used for 0, 1, 3 and 6 consecutive deleted measurements.	38
3.7	Estimated spectral features of CO concentration using the predictive characteristic of Algorithm 1 for $N = 12$ and $N = 24$; (a) a_{0k} , (b) a_{1k} and (c) a_{2k}	39
3.8	Spectral features of CO concentration with missing data for $N = 24$: (a) a_{0k} , (b) a_{1k} , and (c) a_{2k} . “Predictive UFIR” is by Algorithm 1 and “Basic UFIR” is by Algorithm 1 with lines 5–7 removed.	40
3.9	Estimates of temperature on a one-week horizon, $N = 168$, with missing data: (a) temperature measurements and estimates, (b) zoomed part of measurements and estimates, and (c) temperature rate estimates. “Predictive UFIR” is by Algorithm 1 and “Basic UFIR” is by Algorithm 1 with lines 5–7 removed or $\alpha = 1$	42
4.1	Simulated WSN with 50 nodes randomly placed in coordinates of $-40 \text{ m} \leq x \leq 40 \text{ m}$ and $-30 \text{ m} \leq y \leq 30 \text{ m}$: 145 links are due to the node range of 14 m.	54
4.2	Vehicle traveling circularly on a ground space starting with $x_0 = -10 \text{ m}$ and $y_0 = 10 \text{ m}$ at $k = 0$. Unpredictably, an external force affects the movement from $k = 350$ to $k = 370$	56
4.3	Estimation errors produced by nodes with micro-filters along the coordinate x: (a) μKF with the μUFIR filter (1 link) as a benchmark and (b) μUFIR filter with the μKF (1 link) as a benchmark.	57
4.4	RMSEs produced by the nodes with micro filters as functions of the number of the links along the coordinate x: (a) $\delta_k = 0$ and (b) $\delta_k \neq 0$	58

4.5	Effect of the impulsive measurement noise on estimation errors: (a) impulsive noise, (b) 1-link node, and (c) 13-links node.	59
4.6	RMSEs produced by nodes with the μ KF and μ UFIR filter under the unspecified impulsive measurement noise with $q = 1$ and $p > 1$	60
5.1	Simulated WSN with 50 nodes in the area of $-40 \text{ m} \leq (x, y) \leq 40 \text{ m}$ and 144 links (dashed) due to the nodes range of 14 m. The object trajectory (solid) is circular with a sudden maneuver changing the radius.	73
5.2	Estimation errors along axis x produced by dKF under errors in Q_k and $R_k^{(i)}$. The dUFIR filter errors for 1 link are shown as a benchmark. The dH_∞ estimates are shown for two tuning factors $\theta_1 = 0.80768$ and $\theta_2 = 0.80766$ with an extremely small difference.	75
5.3	Estimation errors along axis x produced by the dUFIR filter under errors in Q_k and $R_k^{(i)}$. The dKF errors for 1 link are shown as a benchmark.	76
5.4	Root MSE (RMSEs) as functions of the number of links produced by the dUFIR filter and dKF under $p \neq 1$ and $q \neq 1$. Factor λ_k^{opt} of the dUFIR filter is affected by $q = \{1, 1.2, 1.4, 1.8, 2\}$	77
5.5	A WSN covering a part of the UGV ground truth trajectory (dashed).	78
5.6	Absolute estimation errors along the coordinate x produced by the dKF and dUFIR filter for six scenarios, (SC-1)–(SC-6).	80
5.7	Measured computation time consumed by Algorithm 5 and batch filters (5.15) and (5.20) on different horizons N	82
6.1	An example of a WSN composed of 150 nodes with 687 links (dotted). The WSN covers the moving object trajectory (solid) with the number of the links limited with the nodes ranges.	84
6.2	WSN over a ground truth trajectory available from the MagPIE project dataset [6].	95
6.3	Measurements along coordinate x with missing data for: a) node 10 and b) node 12.	96
6.4	RMSE produced by each node of the WSN.	97

6.5	Estimation error produced by the dUFIR filter and dKF under error in the noise covariances: a) node 12 with 2 links and b) node 10 with 9 links.	98
6.6	A WSN covering the UGV trajectory available from the MagPIE dataset [6]. The labeled nodes are used in the reconstruction of the trajectory. The dashed circle exhibits the 2 m range of a ToF sensor.	100
6.7	Absolute estimation errors along the coordinate x produced by the dKF and dUFIR filter for six scenarios, (SC-1)–(SC-6).	102
6.8	Absolute estimation errors along the coordinate y produced by the dKF and dUFIR filter for six scenarios, (SC-1)–(SC-6).	103
6.9	Estimates provided by the dKF and dUFIR filter along the coordinate y for scenario SC-6.	104
6.10	Simulated network connections of the sensing stations, given their real locations.	105
6.11	Temperature measurements of the eleven sensing stations.	106
6.12	Temperature measurements of the eleven sensing stations.	107
6.13	Temperature measurements and estimates of: a) station 10, b) station 9 and c) station 2.	108
6.14	a) Temperature estimates of all stations and b) disagreement between estimates.	109

List of Tables

4.1	Nodes Sorted by the Number of Links with Neighbors	55
5.1	Nodes Sorted by the Number of Available Links with Neighbors	72
5.2	Nodes Available in Different Time Intervals of Index k	79
5.3	RMSEs Produced by dKF and dUFIR Filter	81
6.1	Nodes Sorted by the Number of Available Links with Neighbors	94
6.2	Nodes Available in Different Time Intervals of Index k	99
6.3	RMSEs Produced by dKF and dUFIR Filter	101

Acronyms

ADC analog-to-digital converter.

ASIC application-specific integrated circuit.

CE consensus on estimates.

CH cluster head.

CI consensus on information.

cKF centralized Kalman filter.

CM consensus on measurements.

cUFIR centralized UFIR.

DSP digital signal processor.

FIR finite impulse response.

FPGA filed-programmable gate array.

GNPG generalized noise power gain.

IIR infinite impulse response.

KF Kalman filter.

MCU microcontroller unit.

MEMS micro-electromechanical systems.

MF multisensor fusion.

NLOS non line of sight.

PDF probability density function.

RF radio frequency.

RH receding horizon.

RMSE root mean squared error.

UFIR unbiased finite impulse response.

WSNs wireless sensor networks.

Chapter 1

Introduction

1.1 Background

1.1.1 Optimum and Robust Estimators

The estimation problem is the process of inferring a determined value of interest, making use of indirect, inaccurate and uncertain observations [7]. The value of interest may be a parameter (a time invariant quantity) or the state of a dynamic system which is time-dependent. A block diagram that illustrates the estimation process is depicted in Fig. 1.1 where the system and measurement errors are inherent of the dynamical and measurement systems, which are caused by mechanical imperfections or environment alterations. Given the stochastic nature of these errors, they are often treated as process and measurement noise, therefore the objective of an estimator would be to *filter out* the noise. This is the reason why an estimator is often referred to as a filter.

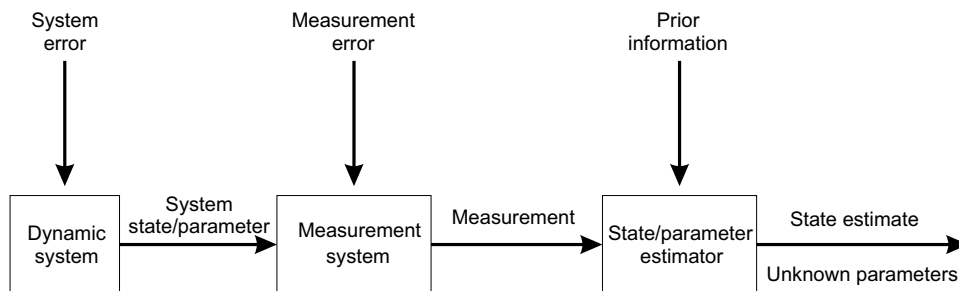


Figure 1.1: Illustration of the estimation process.

Many estimators may be developed depending on very specific criteria, however if an estimator is the best at accomplishing such criteria then it is referred to as an *optimal estimator*, nonetheless some fundamental problems may arise. Given that the optimal estimator is the best, the resulting algorithm may present a high complexity which could render it unsuitable to be implemented in real world applications [8]. Moreover, optimal estimators may require to meet very specific conditions in order to produce optimal results, if such requirements are not guaranteed, errors in the estimates may be too large or even diverge. For these reasons, robustness does not always go along with optimality [9, 10], therefore the need arises to develop estimators that present a desired level of performance against possible system uncertainties, even though optimality is sacrificed. These estimators are called *robust estimators*.

A technology whose applications require robust state estimators are wireless sensor networks (WSNs) [11]. In recent years, WSN technology has permeated into modern society in a variety of applications in industry, healthcare and services [12, 13, 14]. The unique characteristics of WSNs allows for large scale nodes deployments, where redundant measurements of a physical quantity of interest are available and the state estimators must be capable of *fusing* the collected information, in order to reduce the noise in real time. The best noise reduction is typically achieved in WSNs using optimal estimation and fusion techniques [15, 11, 16, 17], which remain in the developments for diverse WSN structures [18, 19, 20, 21].

Wireless sensor networks may be present in three different topologies: centralized, decentralized or distributed. The latter is the most robust and flexible [22, 23, 24], however a *consensus protocol* is mandatory in order to fully exploit its robustness. Olfati-Saber introduced the concept of average consensus in [25] and implemented it in a Kalman filter for consensus on estimates (CE) and consensus on measurements (CM) in [26, 27] respectively and in [28, 29], the consensus on information (CI) was introduced.

The KF is most popular in the design of fusion algorithms [30, 31] mainly due to simplicity, optimality, and low computational burden. In Fig 1.2 the iterative processes of the KF is illustrated, nonetheless it falls behind in terms of robustness due to assumptions in the state model. For the KF to produce optimal results, the following conditions must

be met:

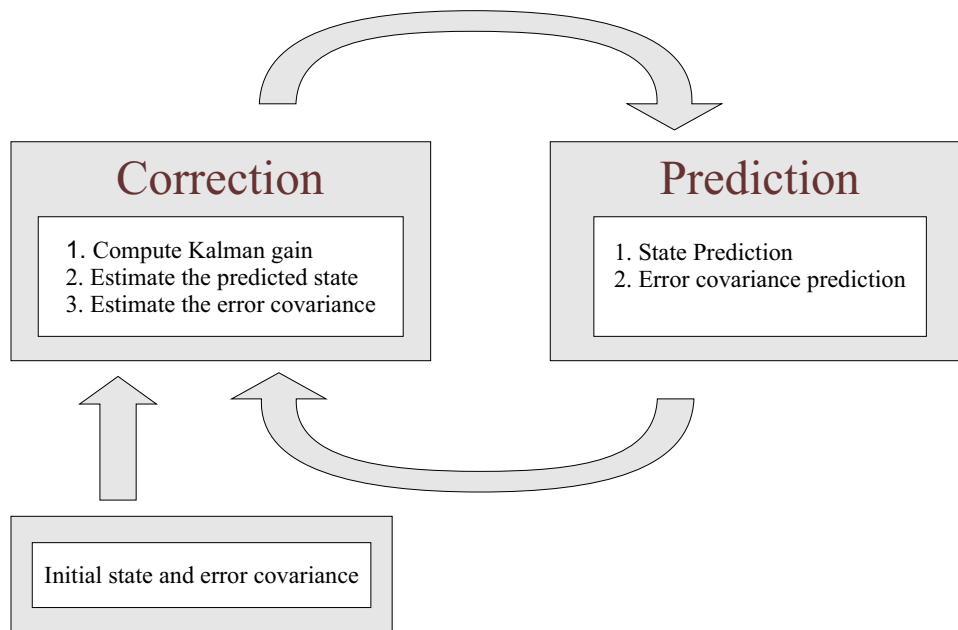


Figure 1.2: Kalman filter iterative process.

- Noise must be white Gaussian.
- The initial conditions are to be known.
- The exact knowledge of the noise statistics is mandatory.

Not guaranteeing any of these conditions causes large errors in the estimates or even divergence of the algorithm. As an example, in Fig 1.3 we illustrate the estimation error of a state variable with and without properly tuning. As expected, the estimation error for the case when optimal conditions are not guaranteed increases.

The restrictions imposed on the Kalman filter makes it unreliable for real life implementations, where accurate system models are not always available. Moreover, the statistical nature of the noise process is often disregarded by the engineers making the fine tuning of the KF a difficult task. For this reason, filters that can handle modeling errors and noise uncertainty were developed. Such is the case of the H_∞ filter[9].

In [9], Simon introduced a Kalman-like algorithm for H_∞ filter based on game theory. As the H_∞ structure is similar to the KF, its implementation is straight forward, except for

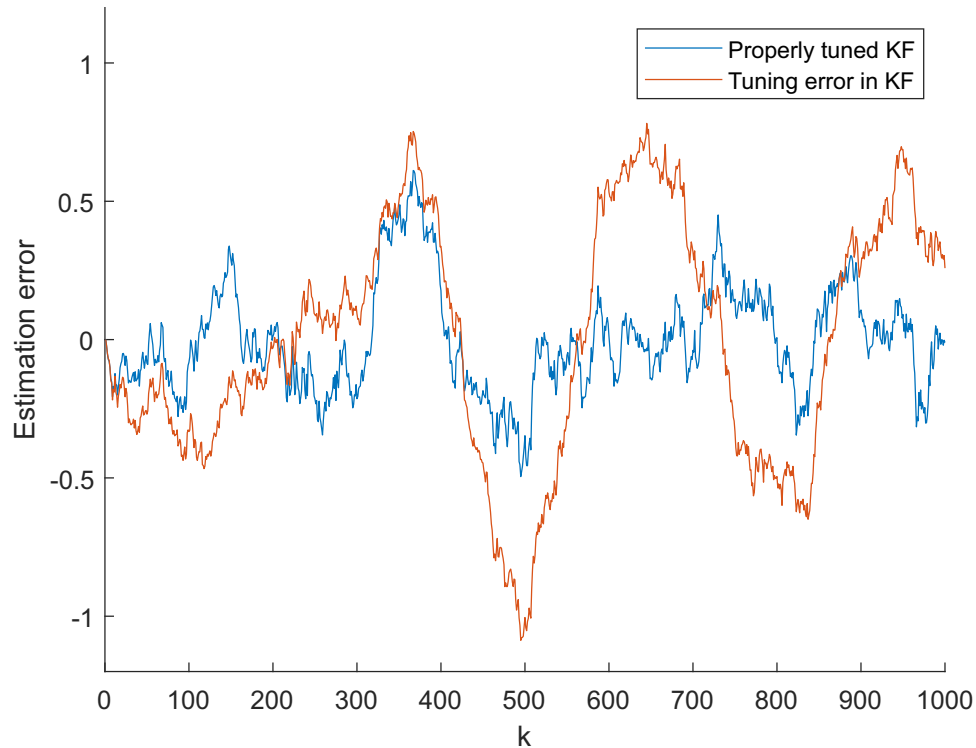


Figure 1.3: Effects of a badly tuned KF.

the user defined parameter θ in the error covariance prediction. If θ is fine tuned, the H_∞ filter presents a more robust behavior than KF. However, if the parameter is not adequate, the error of H_∞ increases considerably. In the example of Fig. 1.4, both behaviors of the H_∞ filter are sketched where a completely unacceptable behavior is observed for choosing a wrong value of θ .

Although the H_∞ filter shows better robustness than KF, the dependency on the appropriate value of θ makes it still an unreliable choice for real life situations. In the previous example the difference between an acceptable and unacceptable behavior in the H_∞ filter is of 0.0505.

Both KF and H_∞ filter present an infinite impulse response (IIR) structure, where past information and errors are used indefinitely for present estimates. This behavior is the principal cause for the lack of robustness in IIR filters. On the other hand, finite impulse response (FIR) filters use limited memory over the most recent time interval [32], which translates in their inherent stability and robustness against temporary uncertainties [33, 34]. The first FIR estimators presented several drawbacks that made it difficult to be

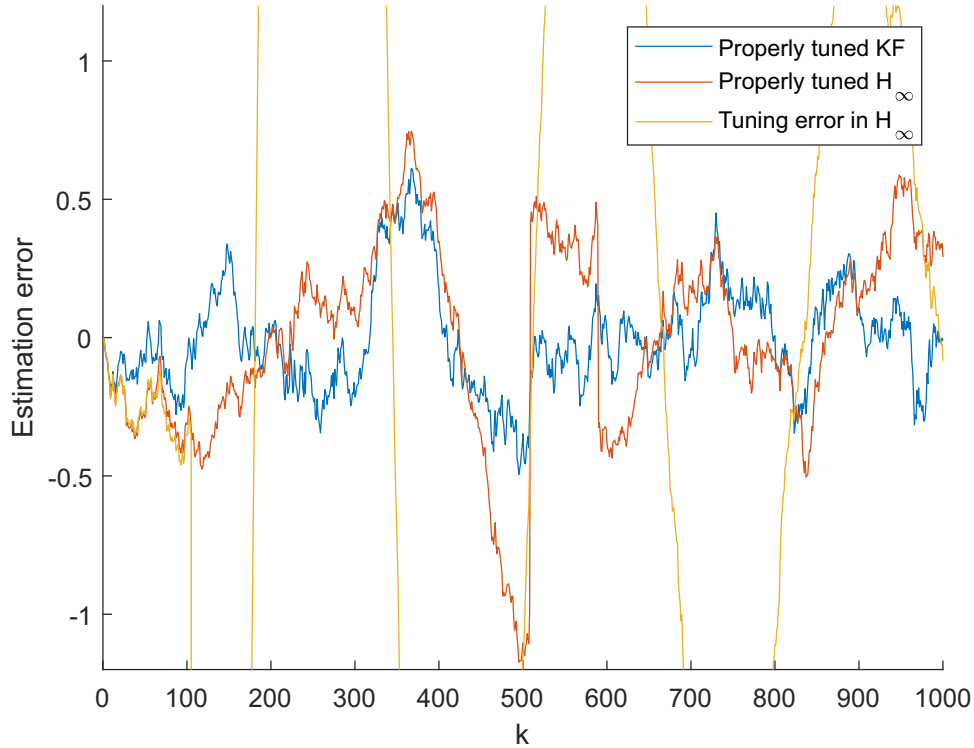


Figure 1.4: Effects of a badly tuned H_∞ .

implemented in real life systems. Firstly, in order to obtain optimal estimates, the noise statistics were to be known and also be Gaussian. Secondly, the lack of a recursive form and limited memory electronics complicated the implementation in real-time systems. Regarding this issues, in [35] an optimal UFIR filter was proposed that did not required the knowledge of noise statistics. The investigation was furthered in [36] to include a polynomial model of the process and in [37] a recursive form, suitable for real-time systems, was presented.

Although the UFIR filter is not optimum, its advantages in terms of robustness, the fact that is completely blind to noise statistics [38] and that the only tuning parameter is its horizon, N ; makes of UFIR filter an attractive alternative to KF and H_∞ . The example in Fig 1.5 corroborates this statement. We observe large errors at the initial estimates due to the horizon, because UFIR filter requires at least N valid measurements. However, after $N + 1$ valid measurements, the filter becomes stable and we observe that the error barely changes in the scenario where KF and H_∞ present large errors, therefore a more robust behavior is achieved by the UFIR filter.

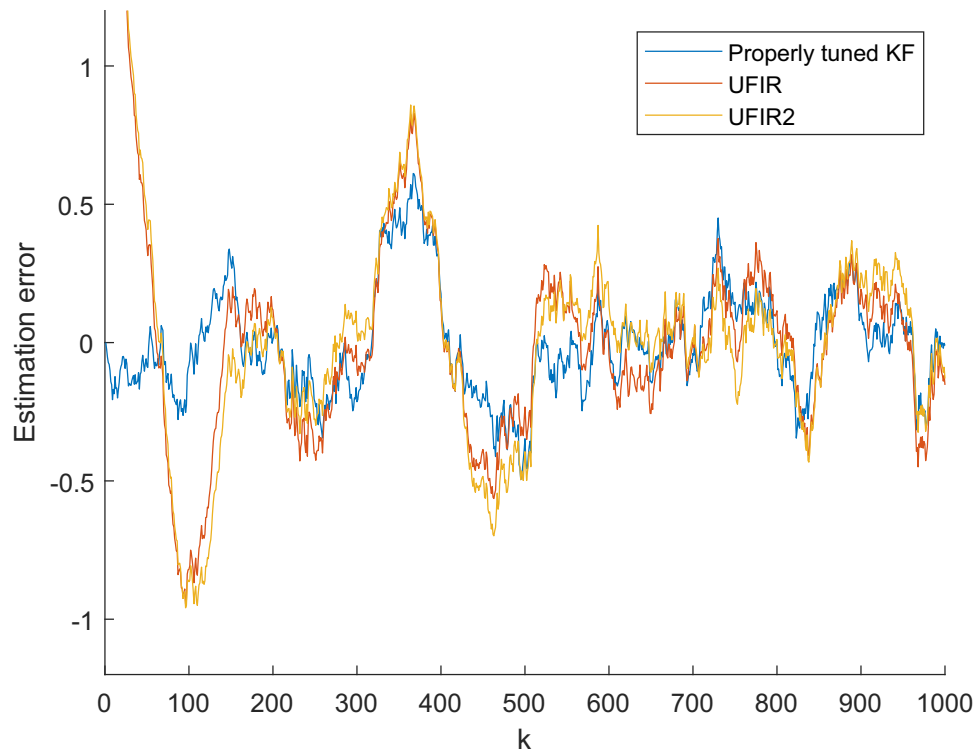


Figure 1.5: Better robustness of UFIR filter.

1.1.2 Multisensor Fusion

In recent years, multisensor fusion has found great acceptance in ubiquitous applications such as industrial machinery monitoring, medical diagnosis, robotics, industrial farming and smart cities [39, 40]. The appeal to this technology comes from the possibility to combine data from multiple sensors and related information from databases in order to improve robustness and achieve better accuracy than using a single sensor. For this reason, multisensor fusion (MF) has become an essential part of applications regarding WSNs. The most straightforward implementation is to centralize the fusion and digital signal processing to a powerful computer (fusion center), so the sensors are only in charge of gathering data and relying it to the fusion center. However, this topology makes the network completely dependent on a single device (the fusion center).

As an alternative to a centralized structure, a distributed approach has been taken. In a distributed estimation scheme, the fusion center is no longer needed, as the entire network performs the estimation and data fusion while reaching a consensus. The consensus term was first used in [41, 26]. Since then, many researchers have proposed new consensus

schemes, which can be classified into four groups: *Consensus on estimates*, *consensus on measurements*, *consensus on information* and H_∞ *consensus*.

The consensus on measurements was introduced in [26] by implementing a micro-KF instead of a centralized filter. The data and error covariance matrices of the neighboring sensors passed through a low pass and band pass consensus filters respectively. The downside of this approach is that the stability can be ensured only when a sufficiently large number of consensus steps are carried out during each sampling interval, so that the local information provided by the innovation pairs has time to spread throughout the whole network. Also, the observation matrices should be equal, however this equality restriction was surpassed in [1] by implementing two High gain high-pass consensus filters. In Fig. 1.6 one can observe how the estimates reach a consensus as the time steps increases.

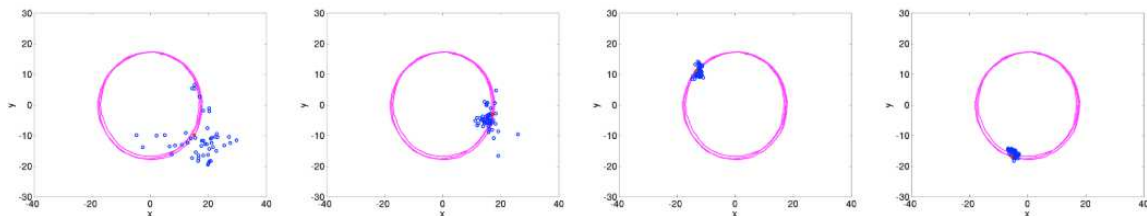


Figure 1.6: Consensus on measurements from [1]

In consensus on estimates, the average of the individual estimates is used to reach a consensus. This implies that the nodes must communicate with its neighbors their respective estimates. This could become an issue if the node must wait for everyone of its neighbors to send their respective estimate. In [1] a Kalman filter based on this type of consensus was proposed. The main idea was to use a two stage KF, one for individual estimations and the other to fuse the individual estimates. Fig. 1.7 shows a better behavior than Fig. 1.6 regarding the consensus.

Consensus on Information was first introduced in [28]. In this work, the authors propose an average of the Kullback-Leibler divergence between the local probability density function (PDF) of the information matrix and information vectors of the neighbors. By assuming that the PDFs belong to the same parametric family, the average minimizes the sum of the information gains. A PDF that produces an information gain as small as possibly is

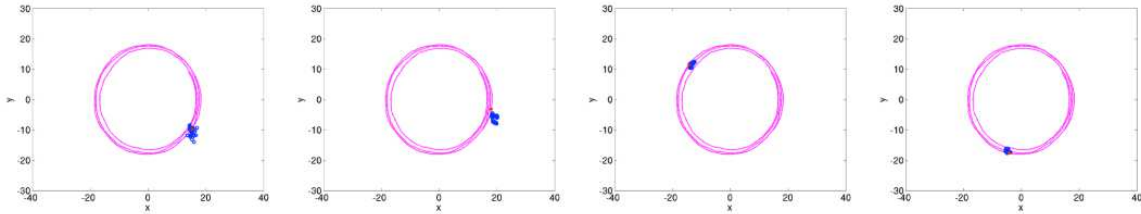


Figure 1.7: Consensus on estimates from [1]

the one that best represents the current state of knowledge. Implementing the consensus on information is straight forward for the information version of the KF [26]. In [2] an unscented KF with consensus on Information is proposed to overcome issues associated to sensor saturations. Partial results are sketched in Fig. 1.8

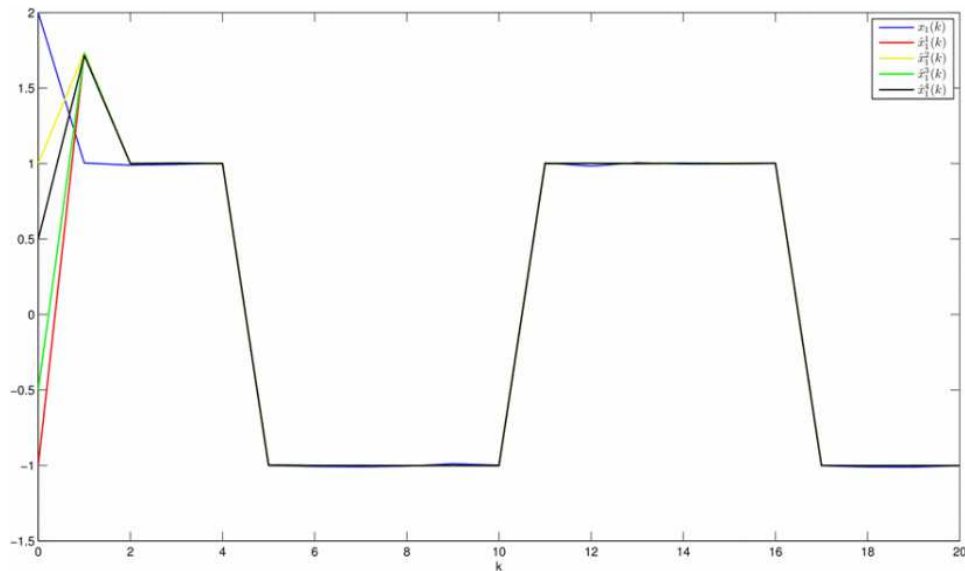


Figure 1.8: Consensus on Information from [2]

As previously discussed, the optimality of KF does not guarantee the robustness needed for real life applications, therefore, H_∞ consensus was proposed. In [3] the first H_∞ distributed consensus was developed to quantify bounded consensus regarding the filtering errors (agreements) over a finite horizon. Partial results can be observed in Fig. 1.9. Although H_∞ algorithms do prove to be more robust than KF, the lack of a standard methodology to define the tuning parameters, puts them at a disadvantage in real life applications, as the designed filter may not remain suitable for unpredicted scenarios, such

as a changing topology due to activation/de-activation of sleeping sensors.

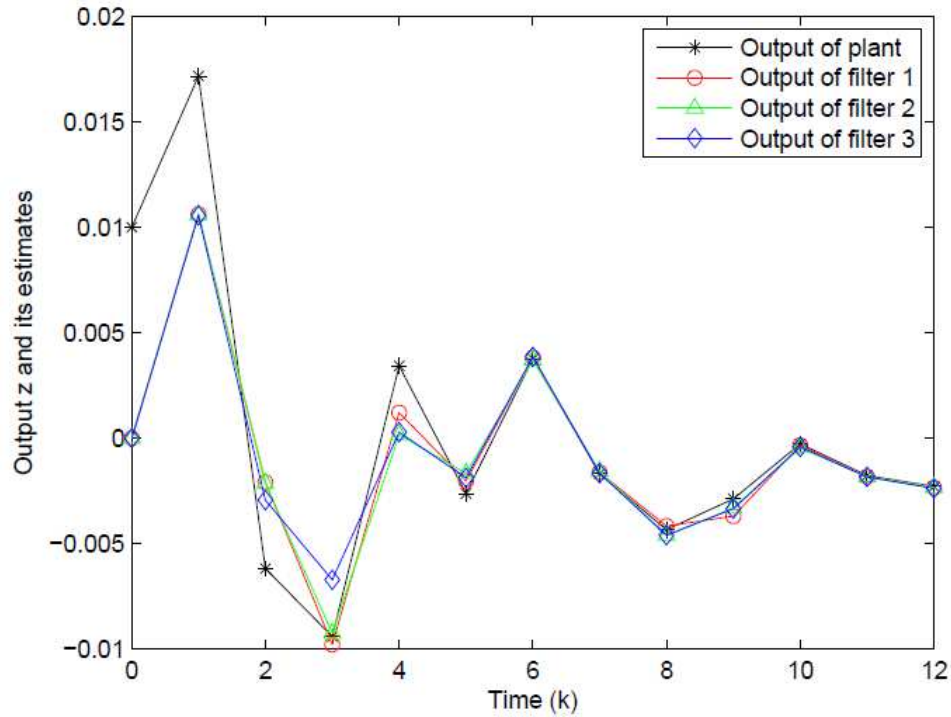


Figure 1.9: H_∞ consensus results from [3]

1.2 Hypothesis

Estimation algorithms based on UFIR filters present less estimation error against miss-model errors, unknown noise statistics and unpredicted model behavior than infinite impulse response filters such as Kalman filter and H_∞ filter. Therefore, UFIR filters with average consensus are better suited for wireless sensor networks in real life applications.

1.3 Motivation

In recent years, wireless sensor networks have been used in many areas, such as environmental and industrial monitoring, localization and tracking and health care. A WSN can be deployed over different regions for a diversity of applications, such as environmental variables monitoring, forest fire detection, carbon monoxide monitoring in urban areas,

unmanned vehicle navigation, among others. Smart sensors, the main units on a WSN, can also be wearable so vital signs can be monitored for health care reasons. All of these applications present unique challenges but, as common ground, the algorithms in charge of state estimation must be sufficiently robust against unexpected model behavior, unknown noise distributions, unexpected alterations in the network, loss of sensor nodes, etc.

It remains a fact that KF is one of the most popular choices for multisensor fusion, mainly because of its easy implementation and optimality. For these reasons, the consensus schemes discussed in the previous section have been mainly developed for KF. There is even an approach based on game theory to develop an H_∞ filter with a FIR structure similar to KF [9]. Despite the wide acceptance of KF, it has been demonstrated that its performance falls behind other technologies in terms of robustness, primarily due to its inherent FIR structure.

Real world applications of WSNs dictates that state estimators must be robust, scalable, energy efficient and fault tolerant [11]. These requirements are not achieved by the KF because, in real world applications, the optimality for the KF is not guaranteed [42, 43]. There are even applications where information about model and sensor noise is unavailable, making the implementation of KF non-viable [38]. Although much effort has been made in the robustification of the KF [44, 45], better robustness is inherent to structures that operate on finite data [46, 47]. In [48] a distributed filter based on a moving horizon approach is presented. In [3], the consensus filtering problem over a finite-horizon for sensor networks with multiple missing measurements is studied and furthered for non-linear systems in [49, 50].

Solutions based on FIR filtering offers several fast algorithms which may efficiently be used in WSNs. A receding horizon (RH) Kalman FIR filter designed in [33] operates similarly to KF on finite horizons. For deterministic time-invariant control systems, a fast recursion-based algorithm was developed in [51]. An iterative p -shift UFIR algorithm proposed in [52] completely ignores the noise statistics and initial values while reducing the output noise variance as a reciprocal of the horizon length. The p -shift UFIR estimator provides filtering with $p = 0$, RH filtering with $p = 1$, $|p|$ -lag smoothing with $p < 0$, and p -step prediction with $p > 0$ [53]. An important feature of the UFIR estimate is that

optimality can be practically achieved on large horizons. Besides, the performance of the UFIR filter can be improved by adapting the generalized noise power gain (GNPG) to operation conditions [54]. Fast optimal FIR algorithms were also designed [47, 55] and some other developments on FIR filtering can be found in [56, 57, 58, 59]. Hence, methods of FIR filtering can be used in design of robust WSNs. However, solutions still have not been addressed to practitioners, making this the motivation for this work.

1.4 Objectives

The principal objective of this thesis is to develop robust distributed unbiased finite impulse response filters, that address the restrictions imposed by the WSNs.

The specific objectives are as follows:

- To develop a fast and memory efficient UFIR filter for its implementation in smart sensors.
- To develop a fast distributed UFIR filter with consensus on measurements.
- To develop a fast distributed UFIR filter with consensus on estimates.
- To demonstrate that the proposed filters are a more robust alternative to the widely used Kalman Filter.

1.5 Scope

In Chapter 2, we present the basics of WSNs and lay down the problem statement. In Chapter 3, a fast recursive Unbiased Finite Impulse Response filter is developed. This filter is suitable to be implemented on a smart sensor given its fast operation and small matrix complexity. The filter is tested with real data of Carbon-monoxide concentration and temperature. In Chapter 4, we engage in a more challenging scenario by considering the unique characteristics of WSNs and developed a UFIR filter with consensus on measurements, which proves better robustness against a similar Kalman filter. In Chapter 5, a consensus on estimates approach was taken to develop a UFIR filter that improves

the RMSE and its robustness against miss-model errors and time varying statistics. In Chapter 6, we further the research presented in Chapter 5 to develop a predictive UFIR filter capable of performing robust estimations against missing data. Finally, in Chapter 7, conclusions are made, along with discussion of future perspectives of the research project.

Chapter 2

Foundations

2.1 Smart Sensors

Advances in micro-electromechanical systems (MEMS) and semiconductor devices have led to the development of low cost transducers and more powerful and smaller micro-processors in such a way that nowadays is possible to find integrated systems called *smart sensors* [60]. The key idea of a smart sensor, is to perform the processing of the signals directly in the sensors, therefore avoiding the need of an external processing unit.

According to the standard IEEE 1451.2 a smart sensor is a device ”*that provides functions beyond those necessary for generating a correct representation of a sensed or controlled quantity. This function typically simplifies the integration of the transducer into applications in a networked environment*” [61, 62]. Under this definition, a smart sensor must be capable of sensing a physical quantity of interest and establish a communication channel with other devices in order to perform a desired action. Despite the specific application for which the smart sensor is to be used, every sensor presents four main components: power unit, sensing unit, communications unit and processing unit as sketched in Fig. 2.1 [63, 11]. Next, we will briefly describe the role of each unit.

2.1.1 Power unit

As shown in Fig. 2.1, the power unit is, without a doubt, the most important part of a smart sensor, since all the other units are dependent on the amount of energy supplied

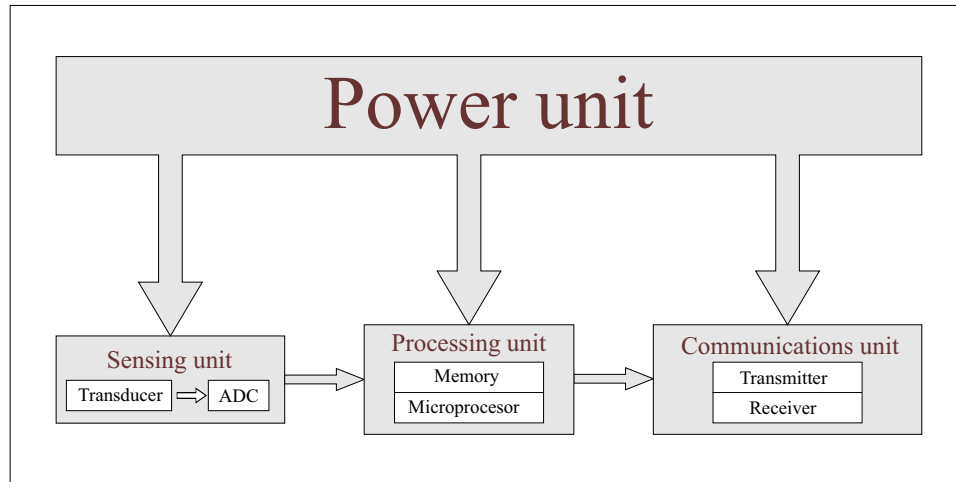


Figure 2.1: Basic representation of a smart sensor.

by this unit. Applications such as environment monitoring require that the smart sensors work as stand alone devices. For this reason, is common practice to equip the sensors with batteries and energy harvesting systems [64]. None the less, once the battery is depleted, the sensor is no longer functional and is disposed off. Depending on the application, the lifetime of a sensor could be from several days to several years, therefore the first and most important design challenge is energy efficiency [65].

2.1.2 Sensing unit

Two of the most important parts of the sensing unit are the transducer and the analog-to-digital converter (ADC). Due to advancement in MEMS and semiconductor technology, nowadays exists transducers capable of transforming a variety of physical quantities into electrical signals which, in turn, will be sampled, discretized, quantized and coded by the ADC. This process is important since it will determine the quality of the measurements. Depending on the transducer, the measurements could present drift, bias, noise, or even non-linearities.

2.1.3 Processing unit

The processing unit is where the *intelligence* of the sensor resides. This unit is in charge of the signal processing according to the specific needs of the application. Nowadays, task such

as decision-making, signal conditioning, noise reduction or even communications among different devices can be carried out by either a microcontroller unit (MCU), digital signal processor (DSP), application-specific integrated circuit (ASIC) or field-programmable gate array (FPGA). The technology to implement is entirely imposed by the application itself. If the sensor can use different transducers, the modularity offer by an MCU would be best, on the other hand if time is an issue, then DSP, FPGA or ASIC will be better suited. Regardless of the technology used, the processing unit must ensure an optimal use of resources, as it will play a key role in the energy consumption of the device.

2.1.4 Communication unit

A smart sensor must be capable of delivering information to other sensors or a decision center. This is done through the communication unit, which includes the transceiver units adequate to the communications channel to be used. Normally, the communication is performed via short-range radio frequencies due to the advantages of an omni-directional transmission and non line of sight (NLOS). However, the radio frequency (RF) technology presents well-known challenges such as saturation of the radio channel, interference, fading, among others which may translate in information loss. The wireless medium is very likely to be shared by multiple network devices, therefore a mechanism is required to control the access to the medium [65]. A key feature of any wireless sensing node is to minimize the power consumption, as the radio subsystem requires the largest amount of power, specially if the sensor has to communicate over long distances[66].

2.2 Wireless Sensor Network

2.2.1 Definition

Wireless Sensor Network is a technology that has been regarded as one of the most important technologies in this century [63]. A WSN consists on a large number of nodes (smart sensors) sharing information through wireless channels about a phenomenon of interest. The rapid grow of WSN is mainly driven by the mass production of cheap and small smart

sensors. The low cost of smart sensors and its small size, makes it possible for a large scale and ubiquitous deployment, which makes them adequate to a large range of challenging tasks from surveillance to health-care [15]. However, the same characteristics that makes them a powerful technology, impose very aggressive constraints that opens up a handful of opportunities for scientific research. The unique characteristics and constraints of WSNs are listed below [63, 15, 67].

- *Large scale deployment.* The low cost of the sensors makes it possible to implement a very dense network with redundancy on wireless connections and measurements which is an advantage in several ways. If for some unpredicted reasons a link is lost, another can be established relatively easy; with redundant measurements, the estimation of the observed phenomenon improves; also by having more nodes per area, the communication distances between nodes will be shorter, implying that the communications unit of the sensor will draw less power from the power unit. However, the larger the network, the more complex its managements. The information may pass through several nodes which would result in unwanted delays and complex routing protocols.
- *Battery-powered nodes.* Several applications require the deployment of nodes in harsh environments where a continuous energy supply is unavailable. For this reason, the implementation of batteries is an adequate solution. However, the restrictions imposed by size and cost will make it unlikely to implement large batteries, affecting directly the life span of the sensor. The limited power supply will automatically demand an efficient power management of all the units in the sensor, therefore the algorithms, communications protocols, and sensing capabilities must take in consideration energy efficiency for once the battery is depleted, the sensor becomes useless.
- *Self-configurable.* Depending on the application, the sensor nodes can be deployed randomly or with prior planning. In a random implementation, the network must establish itself through autonomous configuration of the nodes. In real world applications the network is subject to topology changes due to node addition or failure (node destruction, energy depletion, communication malfunction, etc). For this rea-

son the network must be capable of reconfiguring itself with adequate links in order to keep the communication with all the nodes.

- *No global identification.* The large density of the network makes it impossible for a global addressing scheme, for this would require a very large overhead. This is a challenging issue for event detection applications, where localizing the triggering event is the main purpose of the network.
- *Application specific.* The WSN is designed with an application specific in mind. This implies that for different applications different WSN must be used. Recent MEMS technology makes it possible to have modular sensors which could measure different physical variables, although the burden put on the microprocessor will increase.

2.2.2 Topologies

By definition, a wireless sensor networks consist on several node performing sensing tasks and sending information to a base station or a data sink. Depending on the application, the sink may be a simple gateway to relay information to outside networks or may take a more active approach by sending queries or commands to sensor nodes, perform network management tasks or even apply sensor fusion technics to improve estimates. The way in which the nodes and te sink interact mainly define by the topology of the network. As any other network, the WSN can be found in different topologies, however, due to the their unique characteristics, we can group them in three topologies: star, tree and mesh, which we will discuss below.

Star topology

This topology, depicted in Fig. 2.2, is the less preferred configuration for WSN networks. Non the less if, the number of nodes is small, the network is stable and the nodes present unlimited power supply, the star topology presents unique advantages.

With all nodes sharing their data with the sink, there is no need for routing protocols as the sink will ask for the measurements of the sensors when needed and perform the adequate processing of the data also, with an unlimited power supply, the communication

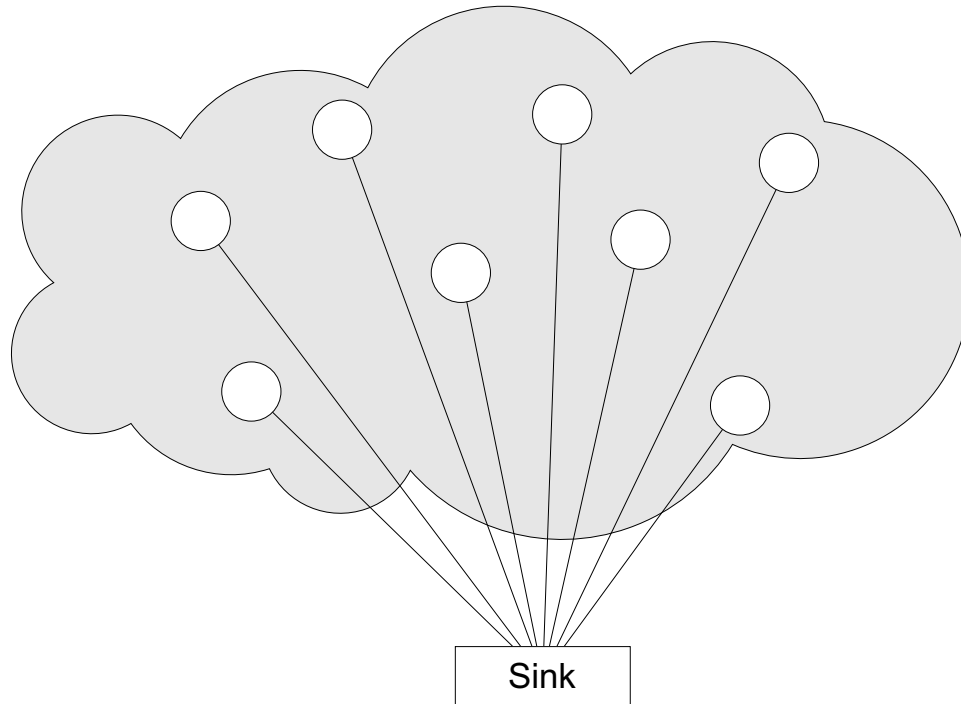


Figure 2.2: WSN presenting a star topology.

distance will only be limited by the transceiver characteristics and the sensing data will be available at all times for the sink to receive it.

It can be seen that the sink must be a very powerful, therefore expensive, device with processing capabilities that outperforms those of the sensor nodes and that the network is completely dependent on this device. Malfunction in the sink will result in the failure of the network. Also, if the number of nodes grows, the sink may be unable of performing all the tasks needed, therefore scalability may not be guaranteed. Lastly, if the nodes are powered with batteries, then the entire network will be useless as the nodes would deplete their batteries in a short amount of time due to long distance communication.

Tree topology

In this topology, the nodes are organized into clusters. Where a cluster head (CH) gathers the data of the cluster members. The CH may perform as a simple gateway to relay the information of the cluster members or may perform data processing such as sensor fusion. The main idea behind this topology is to keep the optimum number of cluster members, therefore the clusters may change depending on the total number of sensors in the network,

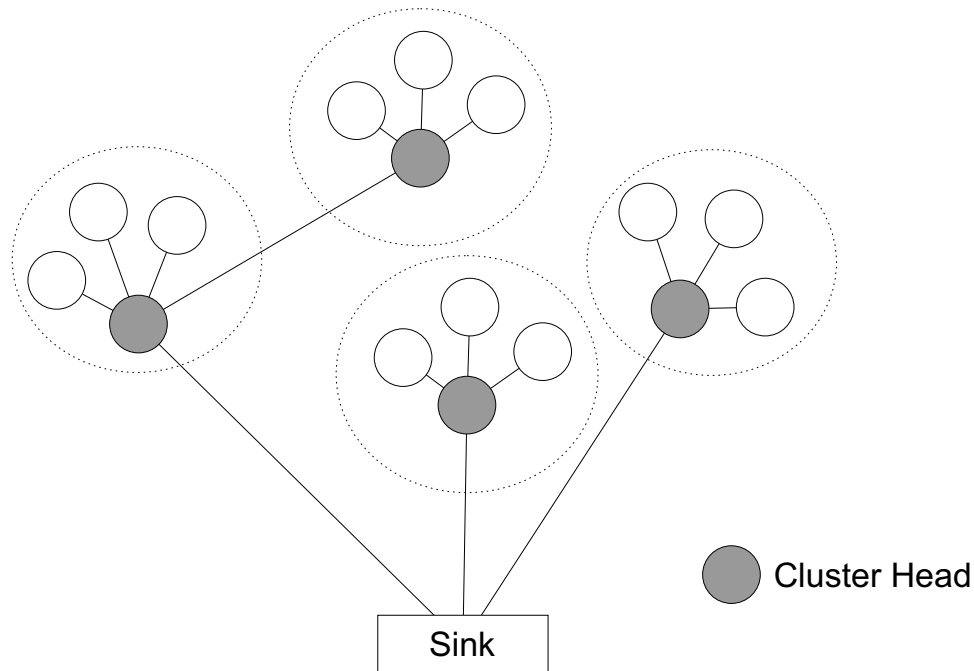


Figure 2.3: WSN presenting a tree topology.

solving the issue of scalability.

None the less, other issues arise. The cluster organization and CH selection are now a mayor concern. Several strategies have been discussed to address this problem [63]. Routing is now an issue, specially when not all the CH are directly connected to the sink as delays may increase, specially in large networks. The CH becomes a very important element. If the CH fails, the entire cluster is compromised, unless a sensor node becomes a new CH.

Mesh topology

A mesh topology is the most robust of the three regarding connectivity. Due to the large and redundant connections, even if some nodes stop functioning it is possible to find other routes to the sink. Also, due to redundancy in the measurements, the estimates will benefit from sensor fusion techniques. However, the lack of special nodes to perform network management or sensor fusion, implies that these functions must be performed by the sensors, meaning that the number of operations for each node increases.

Routing is a mayor problem with this topology as the information could take several

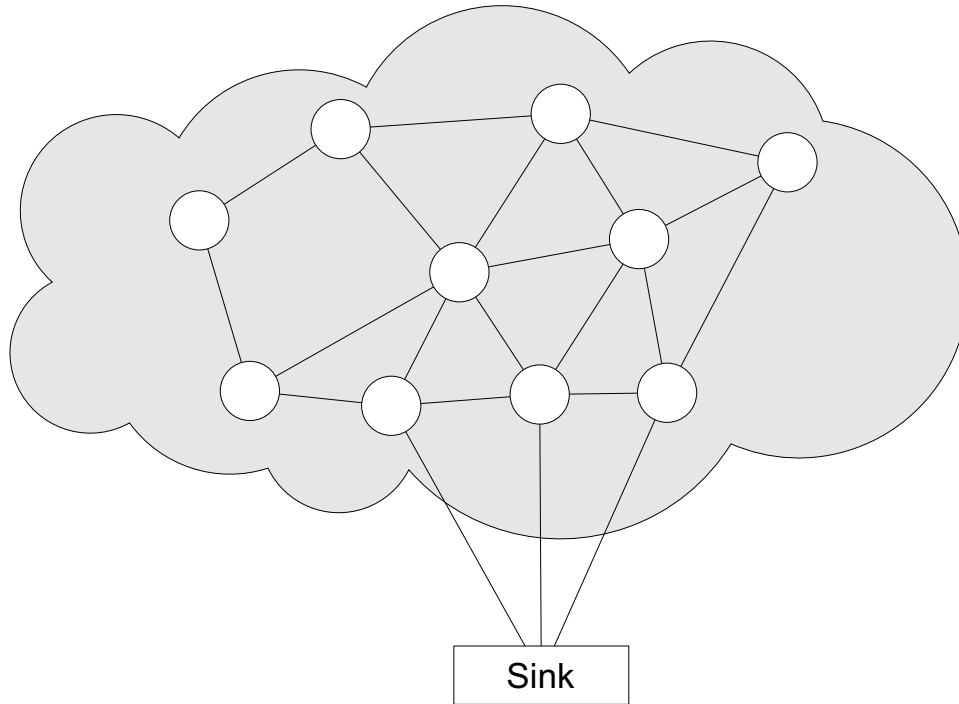


Figure 2.4: WSN presenting a mesh topology.

routes before arriving to the sink node. Furthermore, exist the risk of routing loops formations, resulting in the data never reaching its destiny. One way of avoiding the issues regarding routing is by ensuring that every node in the network reaches a consensus on the estimation of the physical variables. By doing so, the sink will obtain the exact same data regardless of the node. Of course this is only viable for those applications where every node is measuring the same state variables, for example, environmental monitoring.

2.3 Distributed consensus estimation on WSNs

We can identify two main reasons for implementing consensus estimation [16]. The first is due to the lack of a fusion center, which is the case for mesh networks. In this scenario, a consensus protocol will perform the sensor fusion, which in turn will maintain and adequate consensus on the network. The second reason is robustness. The reduced cost of the nodes imply that the sensors are cheap and not very reliable in terms of noise. With consensus estimation, not only a sensor fusion is performed, but also, by reaching a consensus, the overall estimation is improved. Also, the robustness of the network increases given the

possibility that a consensus estimation will be available at every node.

2.3.1 Algebraic graph theory

A WSN can be modeled as an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where each vertex $v^{(i)} \in \mathcal{V}$ is a node for $i \in \mathcal{I} = \{1, \dots, n\}$ and $n = |\mathcal{V}|$. Each link is considered an edge $ij \in \mathcal{E}$. The set of neighbors of node $v^{(i)}$ is denoted by $J^{(i)} = \{j : ij \in \mathcal{E}\}$ and $\deg(v^{(i)}) = |J^{(i)}|$ is called the *degree* of node $v^{(i)}$, which is the total number of first order neighbors for node $v^{(i)}$. The Laplacian of graph \mathcal{G} is

$$L = \mathcal{A} - \Delta, \quad (2.1)$$

where $\Delta = \Delta(\mathcal{G}) = \text{diag}(\deg(v^{(i)}))$ is an $n \times n$ diagonal matrix called *degree matrix* and \mathcal{A} is the *adjacency matrix*. The *Laplacian potential* is defined in [68] as

$$\psi(x) = \frac{1}{2}x^T Lx \quad (2.2)$$

considering $x = (x_1, \dots, x_n)$ as the state vector of the graph \mathcal{G} , which satisfies the following identity

$$x^T Lx = \sum_{ij} (x_i - x_j)^2. \quad (2.3)$$

From 2.3 one can infer that if $x_i = x_j \forall i, j$ then $\psi(x) = 0$, in other words if and only if the value of all the nodes is equal, then the graph potential will be zero. As a result, the Laplacian potential represents the total disagreement of the network, therefore minimizing $\psi(x)$ is the objective of a consensus protocol.

2.3.2 Average consensus protocol

As stated in [25], nodes $v^{(i)}$ and $v^{(j)}$ reach an agreement if and only if the states are related as $x^{(i)} = x^{(j)}$, $\{i, j\} \in \mathcal{I}, i \neq j$. If so, the WSN reaches a consensus with a common value called the *group decision value*. As stated in [69], the most common consensus algorithm for an undirected and strongly connected graph is

$$\psi(x) = -\Delta\psi(x) = -Lx, \quad (2.4)$$

with a stable solution x^* , then $Lx^* = \mathbf{0}$. By the properties of the Laplacian matrix, x^* is the eigenvector associated to the eigenvalue $\lambda = 0$ of L . Since the sum of the row elements in L is zero, it follows that $x^* = [a, \dots, a]^T$, implying that all the nodes in the network reached a common value.

The group decision value is determined by the consensus protocol. In this work, we follow the linear consensus protocol introduced in [68] and formulated as

$$u^{(i)} = \sum_j^J (x^{(j)} - x^{(i)}), \quad (2.5)$$

where $J = |J^{(i)} \cup i|$ is the number of inclusive neighbors. When $u^{(i)} = 0$ an agreement is reached, resulting in $x^{(i)} = x^{(j)} = a = \frac{1}{J} \sum_j^J x^{(j)}$ which is the average of the neighbors initial values [70]. This implies that the network will reach a consensus regardless of the value of the node.

Chapter 3

Design of UFIR Filter for Smart Sensors

As discussed in Chapter 2, the size-cost restrictions of the WSN impose to use cheap and unreliable transducers in the manufacture of smart sensors, causing noisy measurements and data loss. In order to minimize the noise and improve the reliability of the measurements, filtering processes are to be implemented in the sensors. In this chapter, we propose a UFIR filter with prediction capabilities that complies with the unique restrictions of WSNs.

3.1 Introduction

Introduced in [60], the concept of “smart sensor” has been developed and implemented during the following decades [71] to result finally in the IEEE Standard 1451 [62, 72]. A smart sensor is defined as a transducer that generates an electrical signal proportional to a physical, biological, or chemical time-variant quantity $Q(t)$ [62]. In a modern view, a smart sensor is the integration of an analog or digital sensor element, a processing unit, and a communication interface [73, 74]. It is organized as a hardware device with a compact sensing element or actuator. The device typically comprises a microcontroller and communication controller and is supplied with the software to provide signal conditioning, communication, control, calibration, and diagnosis [62].

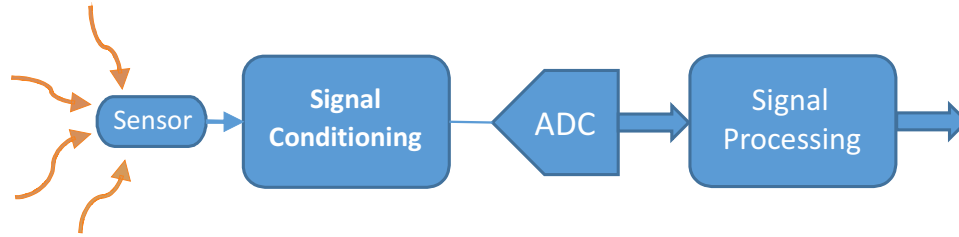


Figure 3.1: Main body of a smart sensor that is responsible for sensing, signal conditioning, ADC, and signal processing. The complete architecture may comprise a DAC, communication units, control units, and an ID tag.

The main body of a smart sensor is responsible for sensing (or multisensing) [75], signal conditioning [76], analog-to-digital conversion (ADC), and signal processing, as shown in Fig. 3.1. A sensitive element converts $Q(t)$ to an electrical signal, signal conditioning is provided to bring an electrical signal into the desired form [77], ADC converts the continuous-time signal into discrete form, and the signal processing unit extracts useful information about $Q(t)$. The complete architecture consists of a digital-to-analog converter (DAC), network communication units, and may be combined with a control (calibration) unit and identification (ID) tag.

Digital signal (information) processing is organized in smart sensors to provide measured data reduction, filtering, signal representation and compression, dynamic error correction, etc. [78, 79]. For a digital filter (estimator) to be practically useful, it must demonstrate [71, 78, 79, 80]:

- *Real-time operation* to avoid delayed data.
- *Unbiasedness* (or *optimality*) with minimum random errors in order to obtain high precision and accuracy.
- *Robustness* against mismodeling, temporary model errors, and errors in the noise statistics.
- *Blind operation* to provide processing with no prior (minimum) information.
- *Predictive features* to estimate quantities with missing data.

Most efficiently, such filters are implemented in state space. Accordingly, we meet several algorithms developed for smart sensors in past decades using the Kalman filter (KF). To cancel the torque ripples in harmonic drive systems, the KF was embedded in [81] to the built-in smart sensor. An intelligent tracking system using KF was designed in [82] by combining several sensors. In [83], the KF was incorporated into a smart sensor in order to estimate the plant state, avoid previous data losses, and substitute the measurement with the KF estimate. The KF was also used in [84] for measurement noise reduction in a multi sensor system. A general structure of a controlled smart sensor has been discussed in [85] to incorporate the KF as a main estimation tool. A low complexity KF based on the differential state equation is used in [86] to provide substantial performance improvements of a smart sensor. A novel architecture was proposed in [87] for a smart micro-electro-mechanical system (MEMS) sensors intended to WiFi fingerprinting by utilizing a KF and an extended KF (EKF). Weighted average and KF were combined in [88] to design voting algorithms for smart sensors.

It has to be remarked now that the KF does not satisfies all the requirements for digital filters in smart sensors. *Optimality* of the KF does not always go along with *robustness*, *scalability*, and *fault tolerance*. Besides, robustness of the KF becomes particularly poor under the imprecisely defined noise statistics [9, 37]. Furthermore, the optimal KF requires all information about the process and measured data and is thus not a blind estimator.

Beyond the KF, several other approaches have attracted researcher's attention. A block recursive adaptive lattice filter was employed in [89] to extract features of the steady-state deviation, frequency, and degree of damping of a power system. The least squares and genetic algorithms were used in [90] to improve dynamic characteristics of smart sensors. An artificial neural network was incorporated in [91] to adapt a capacitive pressure smart sensor to environmental conditions and, in [92], to design an intelligent sensor system for discrimination of material type. Although these methods demonstrate definitive improvements, it has been noticed in [34] that better performance is peculiar to the finite impulse response (FIR) filters [93] operating with most recent data.

Methods of data processing over finite horizons of N data points have attracted attention of designers of wireless sensor networks (WSN) [16, 80] in recent years. In [48], the

concept of moving average was employed to design an estimator under weak observability conditions. A consensus finite-horizon H_∞ filtering algorithm was proposed in [3] for networks with multiple missing measurements. An extension of the finite horizon approach to distributed H_∞ -consensus filtering for time-varying nonlinear networks is given in [49, 50]. In [94], a two-stage recursive structure with norm-bounded parameter uncertainty was designed to force the robust KF to operate on finite horizons similarly to the finite memory approach. In [58], a hybrid particle/FIR filtering structure has been designed to improving reliability of particle filter-based localization in wireless sensor networks.

Let us notice that optimal FIR filtering offers several other fast algorithms which may efficiently be used in smart sensors. A receding horizon (RH) Kalman FIR filter designed in [33] operates similarly to KF. For deterministic time-invariant control systems, a fast recursion-based algorithm was developed in [51]. An iterative p -shift unbiased FIR (UFIR) algorithm proposed in [95, 37] completely ignores the noise statistics and initial values while reducing the output noise variance as a reciprocal of N . This algorithm provides filtering with $p = 0$, RH filtering with $p = 1$, $|p|$ -lag smoothing with $p < 0$, and p -step prediction with $p > 0$. Note that the latter case can be employed to process data with missing measurements [53]. Fast Kalman-like algorithms were designed for optimal UFIR filtering in [47] and for bias-constrained optimal FIR filtering in [55]. An important feature of a simple UFIR filter is that it is blind given N and its estimate becomes practically optimal when the optimal horizon occurs to be large. Besides, the performance of the UFIR filter can be improved by adapting the generalized noise power gain (GNPG) to operation conditions [54]. Hence, methods of fast FIR filtering may open new opportunities in design of smart sensors. However, still no one of these solutions was addresses to designers.

An objective of this chapter is to develop a blind predictive UFIR filtering algorithm operating in smart sensors under missing data with better robustness than in the KF.

3.2 Measured Quantity and Problem Formulation

Let us consider some environment with a quantity $\mathcal{Q}(t)$ (temperature, pressure, location, etc.). Dynamics of $\mathcal{Q}(t)$ is represented with a K -state vector $x_k \in \mathbb{R}^K$ and can be controlled

by a signal $u_k \in \mathbb{R}^K$. A sensing element provides measurements of $\mathcal{Q}(t)$ and, if necessary, the conditioning unit obtains a linearization. Introducing a new filtering technique, we will think that a smart sensor is designed to have a single sensing element, which measures $\mathcal{Q}(t)$ represented with the first state of x_k as a scalar value y_k . We also suppose that measurements can be corrected with a signal e_k to provide calibration or remove some regular trends. Accordingly, the state-space equations for $\mathcal{Q}(t)$ become

$$x_k = A_k x_{k-1} + u_k + B_k w_k, \quad (3.1)$$

$$\tilde{y}_k = H_k (A_k x_{k-1} + u_k), \quad (3.2)$$

$$y_k = \alpha_k (H_k x_k + v_k) + (1 - \alpha_k) \tilde{y}_k + e_k, \quad (3.3)$$

$$z_k = y_k - e_k, \quad (3.4)$$

where $A_k \in \mathbb{R}^{K \times K}$ is the process matrix, $B_k \in \mathbb{R}^{K \times M}$ is the process noise matrix, $H_k \in \mathbb{R}^{1 \times K}$ is the sensor matrix, $w_k \in \mathbb{R}^M$ is the quantity noise, and v_k is the scalar sensor noise. The mutually independent and uncorrelated noise vectors w_k and v_k have zero mean, $E\{w_k\} = 0$ and $E\{v_k\} = 0$, not well-known covariances, $Q_k = E\{w_k w_k^T\}$ and $R_k = E\{v_k v_k^T\}$, and uncertain distributions (not obligatorily Gaussian) that is typical for industrial applications [96]. It is assumed that the ADC is tested for signal quality with a binary output: good ($\alpha_k = 1$) or bad ($\alpha_k = 0$). When $\alpha_k = 0$, the predicted measurement \tilde{y}_k (3.2) is used, in which the known initial x_{k-1} can be substituted with the estimate.

In what follows, $\hat{x}_{k|r}$ is an estimate of x_k at time index k via measurements from past up to and including at time-index r . We will also employ the following variables: $\hat{x}_k^- \triangleq \hat{x}_{k|k-1}$ is the a priori state estimate, $P_k^- \triangleq P_{k|k-1} = E\{(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T\}$ is the a priori estimate covariance, $\hat{x}_k \triangleq \hat{x}_{k|k}$ is the a posteriori state estimate, and $P_k \triangleq P_{k|k} = E\{(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\}$ is the a posteriori error covariance matrix.

Our purpose now is to design a blind predictive Kalman-like UFIR filtering algorithm for (3.1)–(3.4) and show, both analytically and experimentally, that the UFIR filter has higher robustness than the KF. We also wish to test the algorithm proposed by applications implying missing data and under the conditions when the KF cannot be used.

3.3 UFIR Filtering Algorithm

Higher robustness of the UFIR filter [37, 97] is due to its ability to ignore the noise statistics, initial error statistics, and initial values at some (practically insignificant) expanse in accuracy. The only tuning parameter required by the UFIR filter is the optimal horizon length of N_{opt} points, which is applied to minimize the mean square error (MSE) [98, 59]. An important specific of smart sensors is that an average value is typically required over fixed time-intervals (one-hour, one-day, one-week, etc.). For such cases, the UFIR filter provides blind estimation.

The UFIR filter [37, 99] operates with N measured data on a horizon $[m, k]$, from $m = k - N + 1$ to k . Unlike the KF that minimizes the MSE, the UFIR filter satisfies the unbiasedness condition

$$E\{\hat{x}_k\} = E\{x_k\} \quad (3.5)$$

to make the average of the estimate equal to that of the state. The noise variance is reduced here by averaging as a reciprocal of N . Note that filters obeying (3.5) may produce more accuracy than the KF under uncertain conditions [37].

In the discrete convolution form, the UFIR estimate can be found on $[m, k]$ for $\alpha_k = 1$ as [100, 52]

$$\hat{x}_k = \mathcal{H}_{m,k}(Z_{m,k} - L_{m,k}U_{m,k}) + S_{m,k}U_{m,k}, \quad (3.6)$$

where $U_{m,k} = [u_m^T \ u_{m+1}^T \ \dots \ u_k^T]^T$ and the homogenous UFIR filter gain is

$$\mathcal{H}_{m,k} = (C_{m,k}^T C_{m,k})^{-1} C_{m,k}^T, \quad (3.7a)$$

$$= G_k C_{m,k}^T, \quad (3.7b)$$

where $G_k = (C_{m,k}^T C_{m,k})^{-1}$ is called the generalized noise power gain (GNPG) [37]. The

extended observation vector $Z_{m,k}$ and mapping matrix $C_{m,k}$ can be written as

$$Z_{m,k} = [y_m - e_k \quad y_{m+1} - e_{m+1} \quad \dots \quad y_k - e_k]^T, \quad (3.8)$$

$$C_{m,k} = \begin{bmatrix} H_m(\mathcal{F}_k^{m+1})^{-1} \\ H_{m+1}(\mathcal{F}_k^{m+2})^{-1} \\ \vdots \\ H_{k-1}A_k^{-1} \\ H_k \end{bmatrix}, \quad (3.9)$$

where the product of system matrices is defined by

$$\mathcal{F}_k^r = \begin{cases} A_k A_{k-1} \dots A_r, & r < k+1 \\ I & r = k+1 \\ 0 & r > k+1 \end{cases}, \quad (3.10)$$

and auxiliary matrices are given by

$$S_{m,k} = \left[\underbrace{\mathcal{F}_k^{m+1} \quad \mathcal{F}_k^{m+2} \quad \dots \quad A_k \quad I}_N \right], \quad (3.11)$$

$$L_{m,k} = \bar{C}_{m,k} S_{m,k}, \quad (3.12)$$

$$\bar{C}_{m,k} = \text{diag} \left(\underbrace{H_m \quad H_{m+1} \quad \dots \quad H_k}_N \right). \quad (3.13)$$

The batch form (3.7a) may not be suitable for smart sensors in view of the computational complexity. Therefore, we next proceed with the fast predictive iterative algorithm.

3.3.1 Predictive Iterative UFIR Filtering Algorithm

Following the UFIR filter strategy [37], the estimate at k can be obtained iteratively using an auxiliary variable l beginning with $l = m + K$ and ending when $l = k$. To run iterations, the initial estimate at $l = m + K - 1$ can be found using (3.7b) in a short batch form on a horizon $[m, m + K - 1]$, because the inverse in (3.7a) does not exist otherwise.

Provided G_{m+K-1} and \hat{x}_{m+K-1} , the estimate \hat{x}_k is obtained iteratively for l increasing

from $m + K$ to $l = k$ as

$$G_l = [H_l^T H_l + (A_l G_{l-1} A_l^T)^{-1}]^{-1}, \quad (3.14)$$

$$K_l = G_l H_l^T, \quad (3.15)$$

$$\hat{x}_l^- = A_l \hat{x}_{l-1} + u_k, \quad (3.16)$$

$$y_l = \begin{cases} y_l, & \alpha_l = 1 \\ H_l \hat{x}_l^-, & \alpha_l = 0 \end{cases}, \quad (3.17)$$

$$\hat{x}_l = \hat{x}_l^- + K_l (y_l - e_k - H_l \hat{x}_l^-). \quad (3.18)$$

A pseudo code of this algorithm is given as Algorithm 1. The algorithm readily simplifies

Algorithm 1: Predictive Iterative UFIR Algorithm

Data: y_k, e_k, u_k, α_k

Result: \hat{x}_k

1 **begin**

2 **for** $k = N - 1 : \infty$ **do**

3 $m = k - N + 1, \quad s = m + K - 1;$

4 $G_s = (C_{m,s}^T C_{m,s})^{-1};$

5 **if** $\alpha_k = 0$ **then**

6 $y_k = H_k (A_k \hat{x}_{k-1} + u_k);$

7 **end if**

8 $\tilde{x}_s = G_s C_{m,s}^T Z_{m,s};$

9 **for** $l = s + 1 : k$ **do**

10 $G_l = [H_l^T H_l + (A_l G_{l-1} A_l^T)^{-1}]^{-1};$

11 $\tilde{x}_l^- = A_l \tilde{x}_{l-1} + u_l \quad \tilde{x}_l = \tilde{x}_l^- + G_l H_l^T (y_l - e_l - H_l \tilde{x}_l^-);$

12 **end for**

13 $\hat{x}_k = \tilde{x}_k;$

14 **end for**

15 **end**

16 † First data y_0, y_1, \dots, y_{N-1} must be available.

to the homogenous case, by setting $u_k = 0$ and $e_k = 0$. As can be seen, Algorithm 1 does

not require neither the noise statistics nor the initial values that is an important advantage against the KF.

Only one variable N_{opt} is required by the UFIR filter that can be found at a test stage by minimizing the trace of the estimation error P_k as

$$N_{\text{opt}} = \arg \min_N \{ \text{tr } P_k(N) \}, \quad (3.19)$$

by utilizing observations with no reference as shown in [98], or utilizing an advanced technique [59]. Provided N , the UFIR filter becomes a blind estimator.

An important feature of Algorithm 1 is an ability to produce suboptimal estimates under temporary missing measurement data [53]. Such a situation may occur when a sensing element temporarily responds to $\mathcal{Q}(t)$ with incorrect values and a signal conditioning unit is unable to adjust data that is indicated with $\alpha = 0$. If so, Algorithm 1 (lines 5–7) bridges a gap over temporary lost data by predicting y_k via prior estimate, as in (3.17). To take advantage of the prediction mode, y_k must be available on a horizon of N initial points. Otherwise, the estimate \hat{x}_{N-1} must be provided.

3.4 Robustness of UFIR Filter and KF

The trade-off in robustness between the UFIR filter and KF (Algorithm 2) can be learned if to trace effect of model errors and disturbances on the GNPG G_k and Kalman gain K_k . Kalman filtering suggests that an estimator will produce more random errors if the bias correction gain exceeds an optimal value and more bias errors otherwise. In what follows, we will refer to this rule.

3.4.1 Imprecisely Defined Noise Statistics

Let us suppose that the noise covariances, Q_k and R_k , are defined imprecisely and substitute $Q_k \leftarrow \alpha^2 Q_k$ and $R_k \leftarrow \beta^2 R_k$, where $\alpha = 1$ and $\beta = 1$ stand for completely known noise. The UFIR filter ignores noise, thus α and β do not affect its performance. To learn effect of α and β on the KF, we consider a stationary mode, set $P_{k-1} \cong P_k^-$ in line 4 of Algorithm

Algorithm 2: Recursive KF Algorithm**Data:** \hat{x}_0, P_0, Q_k, R_k **Result:** \hat{x}_k, P_k

```

1 begin
2   for  $k = 1, 2, \dots$  do
3      $\hat{x}_k^- = A_k \hat{x}_{k-1} + u_k$  ;
4      $P_k^- = A_k P_{k-1} A_k^T + B_k Q_k B_k^T$  ;
5      $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$  ;
6      $\hat{x}_k = \hat{x}_k^- + K_k (y_k - e_k - H_k \hat{x}_k^-)$  ;
7      $P_k = (I - K_k H_k) P_k^-$  ;
8   end for
9 end

```

2, and transform it to the discrete-time algebraic Lyapunov equation [101]

$$P_k^- - A_k P_k^- A_k^T = \alpha^2 B_k Q_k B_k^T, \quad (3.20)$$

which solution is known to be an infinite sum [102],

$$P_k^- = \alpha^2 \sum_{i=0}^{\infty} A_k^i B_k Q_k B_k^T (A_k^i)^T = \alpha^2 \Sigma_k. \quad (3.21)$$

We next substitute (3.19) into line 5 of Algorithm 2 and write the Kalman gain as

$$\begin{aligned} \bar{K}_k &= \alpha^2 \Sigma_k H_k^T (H_k \alpha^2 \Sigma_k H_k^T + \beta^2 R_k)^{-1} \\ &= \Sigma_k H_k^T \left(H_k \Sigma_k H_k^T + \frac{\beta^2}{\alpha^2} R_k \right)^{-1}. \end{aligned} \quad (3.22)$$

The optimal value $\bar{K}_k = K_k$ is guaranteed by $\alpha = \beta = 1$ [in (3.21), the optimality is also achieved with $\alpha = \beta$]. If $\bar{K}_k > K_k$, the random errors dominate and, if $\bar{K}_k < K_k$, the bias errors grow. That means that 1) random errors will grow if $\alpha > 1$ and/or $\beta < 1$; 2) bias errors will dominate when $\alpha < 1$ and/or $\beta > 1$; and 3) errors may compensate by $(\alpha, \beta) < 1$ or $(\alpha, \beta) > 1$.

Thus, the KF is not protected against errors in the noise covariances. The KF will retain supremacy when α and β are both range close to unity. Otherwise, the UFIR filter that is robust to changes in α and β may be more accurate.

3.4.2 Temporary Model Errors

To learn effect of temporary model errors, we allow $A_k \leftarrow \eta A_k$ with $\eta \neq 1$ during a short time and retain $Q_k \leftarrow \alpha^2 Q_k$ and $R_k \leftarrow \beta^2 R_k$. Because η affects both filters, we rewrite K_k and G_l as

$$\bar{K}_k = \Sigma_k H_k^T \left(H_k \Sigma_k H_k^T + \frac{\beta^2}{\alpha^2 \eta^2} R_k \right)^{-1}, \quad (3.23)$$

$$\bar{G}_l = \left[H_l^T H_l + \frac{1}{\eta^2} (A_l G_{l-1} A_l^T)^{-1} \right]^{-1}. \quad (3.24)$$

It follows that, by $\alpha = \beta = 1$, robustness of both filters to changes in η is near equal. Otherwise, α and β may dramatically increase errors caused by η in the KF. We do not see such an effect in the UFIR filter.

3.4.3 Temporary Measurement Errors

To model measurement errors, we substitute H_k with μH_k , where $\mu = 1$ denotes accurate measurements, and retain $Q_k \leftarrow \alpha^2 Q_k$ and $R_k \leftarrow \beta^2 R_k$. We then transform K_k and G_l to

$$\bar{K}_k = \frac{1}{\mu} \Sigma_k H_k^T \left(H_k \Sigma_k H_k^T + \frac{\beta^2}{\alpha^2 \mu^2} R_k \right)^{-1}, \quad (3.25)$$

$$\bar{G}_l = \frac{1}{\mu^2} \left[H_l^T H_l + \frac{1}{\mu^2} (A_l G_{l-1} A_l^T)^{-1} \right]^{-1} \quad (3.26)$$

and infer the following. When the second components dominate in the parentheses of (3.25) and brackets of (3.26), the UFIR filter has higher robustness, because the gain $1/\mu^2$ is better compensated by the reciprocal of $1/\mu^2$ in (3.26) than $1/\mu$ in (3.25). Moreover, α and β may dramatically deteriorate the KF performance that also speaks in favor of the UFIR filter.

An overall conclusion that can be made based on this analysis is that the UFIR filter is more robust than the KF in real-world applications. One finds evidences of this statement in many papers [9, 34, 33, 47, 55, 103, 104, 105].

3.5 Experimental Verification

Like the KF, the iterative UFIR filter is universal for linear state-space models. In this section, we give two examples of its practical applications to air pollution monitoring and temperature measurements. Although different kinds of air pollution components are of importance for human life, we focus only on carbon monoxide (CO). We base these examples on measurements provided in [4] and available from [5]. The multi-sensor device (electronic nose) used in [4] is developed by Pirelli Labs using seven solid-state sensors. Reference data were provided in [4] using a Conventional air pollution monitoring station.

3.5.1 CO Concentration

CO is colorless, odorless and tasteless gas that is slightly less dense than air [106]. Its concentration nominal range is 0.1–100 mg/m³. But it is toxic to humans when encountered in concentrations above about 40 mg/m³ and kills in an hour by 10³ mg/m³ [106]. It is known that an increase in the level of CO reduces the amount of oxygen carried by hemoglobin around the human body. The amount of oxygen also diminishes in vital organs, such as the brain, nervous tissues and the heart, and they may not work properly [107]. Therefore, monitoring of CO concentration is desirable in urban and industrial areas.

Incomplete combustion regardless of the fuel used and operation of diverse devices result in the CO presence in the air. Therefore, sensors of CO can also be used as indicators of possible fire. Figure 3.2 illustrates two sources of increased CO concentration in air: industrial pollution and car pollution.

State-Space Representation

In order to demonstrate efficiency of Algorithm 1, we employ a 20-days part of long-term CO concentration measurements provided in an urban area [5]. Each point is obtained here by one-hour averaging with no missing data. To convert the sensor output z_k to the reference scale, we use a linear regression and represent converted measurement y_k via actual one z_k as $y_k = -5.8 + 6.5 \times 10^{-3} z_k$. The output y_k (Sensor) and reference z_k (Ref) are shown in Fig. 3.3. As can be seen, the sensor is almost as accurate as the reference source.

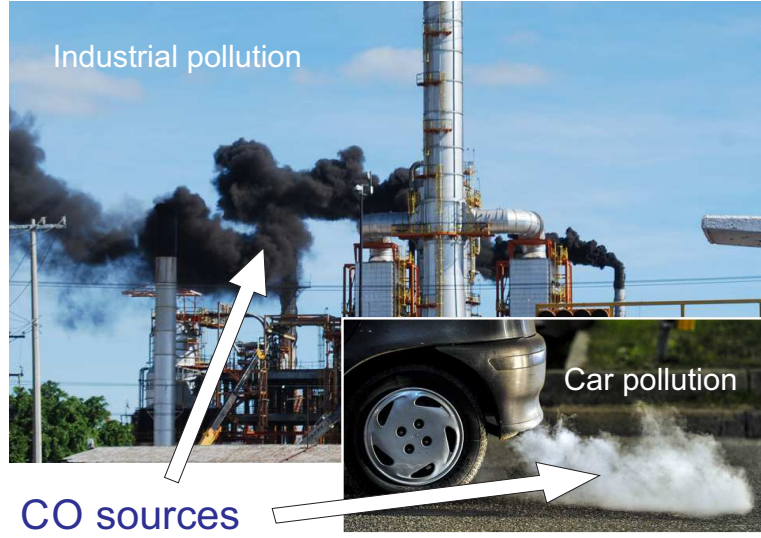


Figure 3.2: Two sources of increased carbon monoxide concentration in air: industrial pollution and car pollution.

It can also be observed in Fig. 3.3 that the CO concentration has 24-hours periodicity and can be considered to be a stochastic and quasi stationary process on a long baseline. The concentration of the day demonstrates two maxima associated with peak hours, a deep minimum corresponding to night time, and a minimum at the middle of the day. We thus represent measurement y_k on a horizon of N points with the second-order Fourier series as

$$y_k = a_{0k} + a_{1k} \cos\left(\frac{2\pi}{24}k + \phi_1\right) + a_{2k} \cos\left(\frac{4\pi}{24}k + \phi_2\right) + v_k, \quad (3.27)$$

where v_k is the zero mean measurement noise and a_{0k} , a_{1k} , and a_{2k} are time-varying spectral features of CO concentration to be estimated by a smart sensor. Measurements in Fig. 3.3 also suggest that $\phi_1 = \pi/5.21$ and $\phi_2 = 2\phi_1$. Note that ϕ_k can be set to zero by removing some initial data points.

In view of quasi stationarity of CO concentration and completely unknown process noise, we assign the state vector $x_k = [a_{0k} \ a_{1k} \ a_{2k}]^T$, allow $x_k = x_{k-1}$ by $A_k = I$, ignore noise by $w_k = 0$ and $Q_k = 0$, and represent y_k with

$$y_k = H_k x_k + v_k, \quad (3.28)$$

where

$$H_k = \left[1 \quad \cos\left(\frac{2\pi}{24}k + \phi_1\right) \quad \cos\left(\frac{4\pi}{24}k + \phi_2\right) \right]. \quad (3.29)$$

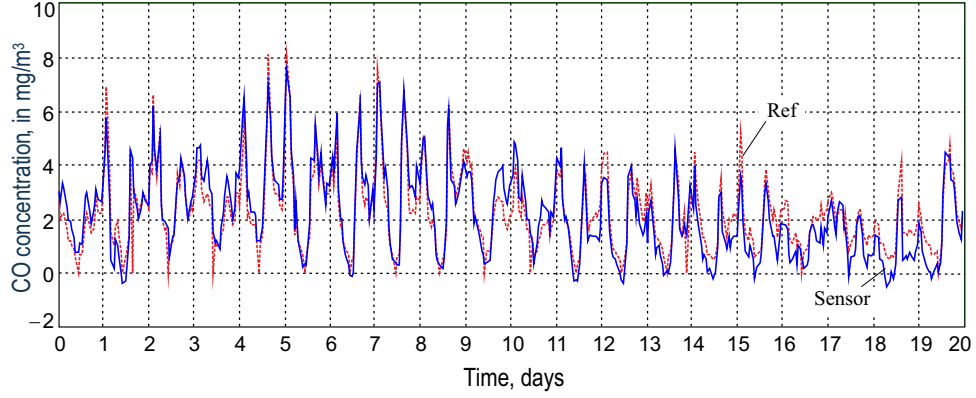


Figure 3.3: 20 days reference (Ref) and real (Sensor) measurements of carbon monoxide concentration in part of mg/m^3 [4, 5].

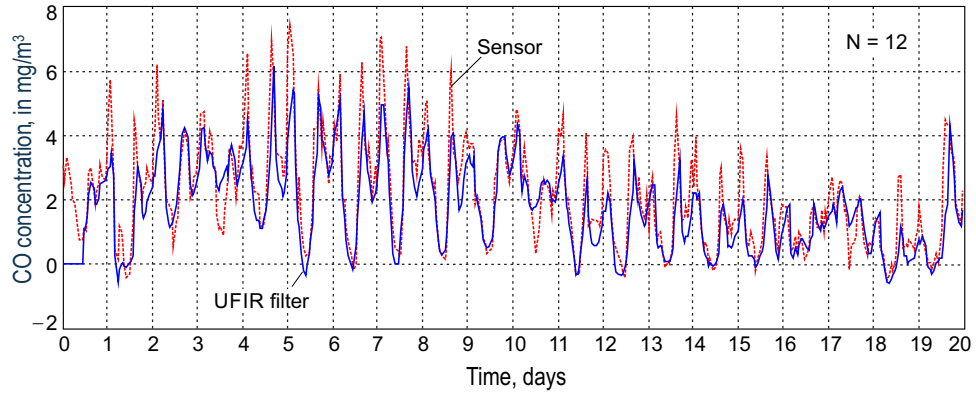


Figure 3.4: CO concentration measured by a sensor (Sensor) and estimated using Algorithm 1 (UFIR filter) on a horizon of $N = 12$ points. Data at each point are one-hour averaged.

For $Q_k = 0$, the KF cannot be used. To exploit the UFIR Algorithm 1, we define an extended matrix $C_{m,s}$ as

$$C_{m,s} = \begin{bmatrix} 1 & \cos(\psi_2 + \phi_1) & \cos(2\psi_2 + \phi_2) \\ 1 & \cos(\psi_1 + \phi_1) & \cos(2\psi_1 + \phi_2) \\ 1 & \cos(\psi_0 + \phi_1) & \cos(2\psi_0 + \phi_2) \end{bmatrix}. \quad (3.30)$$

where $\psi_0 = \frac{2\pi s}{24}$, $\psi_1 = \frac{2\pi(s-1)}{24}$, and $\psi_2 = \frac{2\pi(s-2)}{24}$.

Using Algorithm 1, we first estimate CO concentration as shown in Fig. 3.4, by substituting spectral features in (3.27) with their estimates \hat{a}_{0k} , \hat{a}_{1k} , and \hat{a}_{2k} . Observing this figure, one may conclude that a half-day horizon, $N = 12$, allows for some noise reduction and smoothing regular daily variations. We next extract spectral features of CO

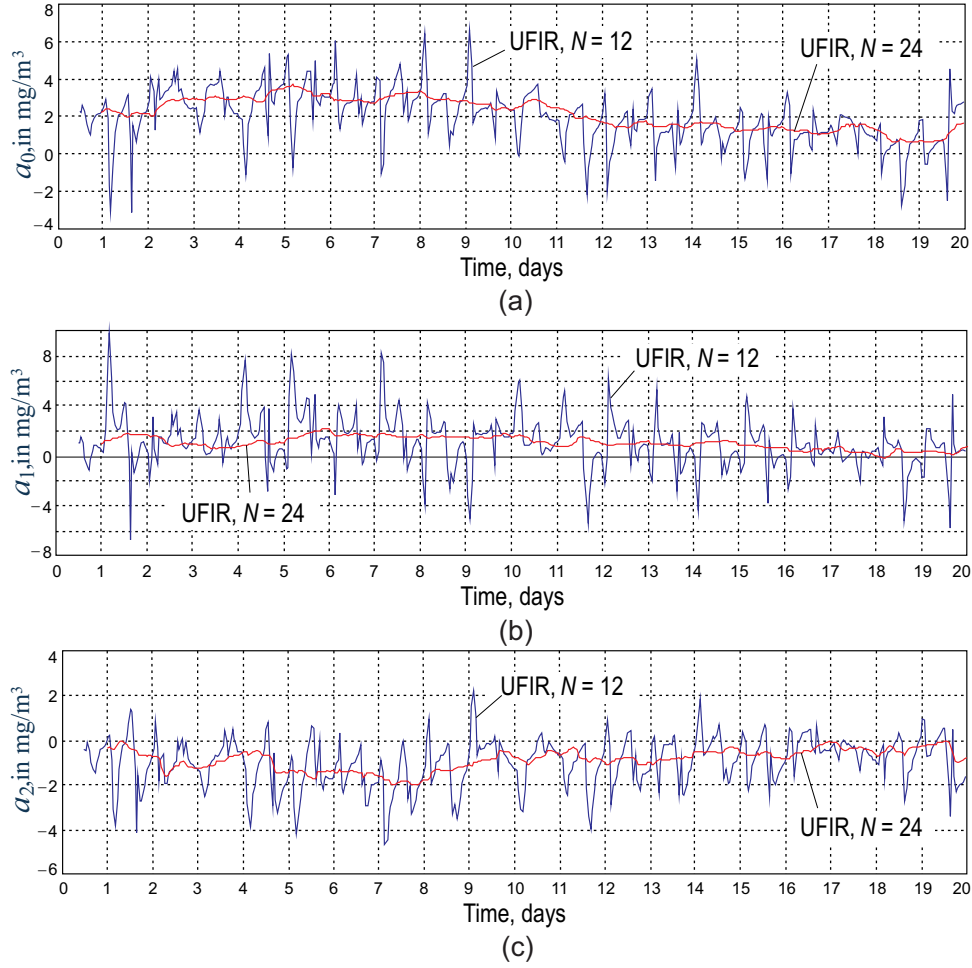


Figure 3.5: Spectral features of CO concentration estimated using Algorithm 1 with $N = 12$ and $N = 24$: (a) a_{0k} , (b) a_{1k} , and (c) a_{2k} .

concentration and analyze their functions in more detail.

Spectral Features of CO Concentration

Values of a_{0k} , a_{1k} , and a_{2k} can be estimated by Algorithm 1 straightforwardly via $\hat{x}_k = [\hat{a}_{0k} \hat{a}_{1k} \hat{a}_{2k}]$. Figure 3.5 sketches the relevant estimates obtained on horizons of $N = 12$ and $N = 24$. It follows that a half-day horizon, $N = 12$, is more suitable for the estimation of fast variations, while $N = 24$ gives well-smoothed estimates. When CO concentration is low, then $N = 24$ can be applied to avoid false alarm. Otherwise, $N = 12$ must be applied. The choice of N is definitely a matter of optimization in view of the fact that high CO concentration can be dangerous only if it is accumulated during some time. Thus, short-

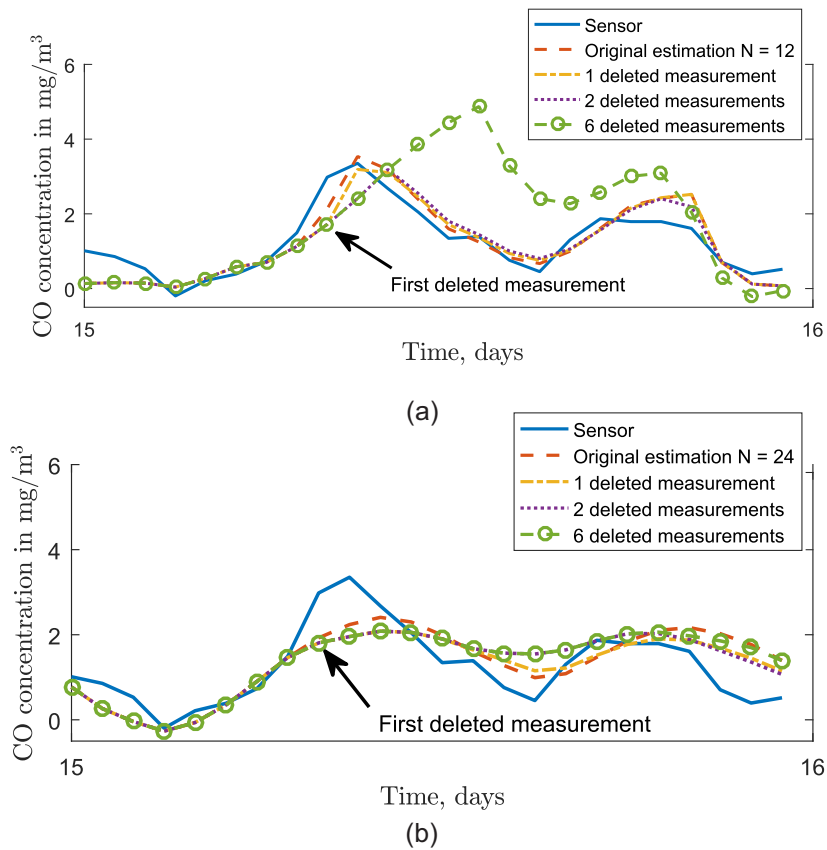


Figure 3.6: CO concentration measured by a sensor (Sensor) and estimated using the prediction feature of Algorithm 1. The horizon (a) $N = 12$ and (b) $N = 24$ are used for 0, 1, 3 and 6 consecutive deleted measurements.

time excursions may not produce essential effect and can be smoothed, while long-time ones must be detected.

Missing Measurement Data

In order to establish the error performance algorithm 1, part of the data from day 15 to 16 was intentionally deleted. Fig. 3.6a and b shows the behavior of the UFIR algorithm with $N = 12$ and $N = 24$ respectively, for one, three and six consecutive missing measurements.

In Fig. 3.6a, it can be seen that for an small amount of lost data (less than three deleted measurements), the difference between the measured and estimated CO concentration is relatively small. A larger difference becomes apparent as the amount of missing data increases. On the other hand, for the horizon $N = 24$ (Fig. 3.6b), the estimates are less

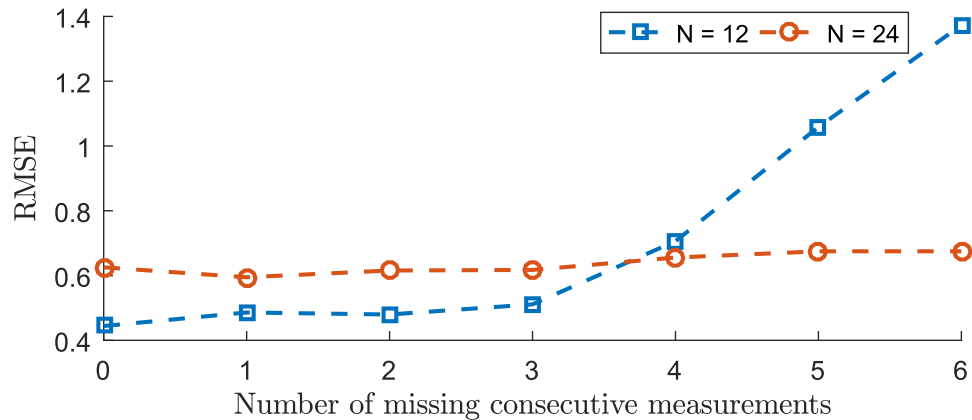


Figure 3.7: Estimated spectral features of CO concentration using the predictive characteristic of Algorithm 1 for $N = 12$ and $N = 24$; (a) a_{0k} , (b) a_{1k} and (c) a_{2k} .

affected for the same amount of missing data, as can be appreciated on. This is further corroborated by the root mean squared error, calculated over the time span of analysis, and sketched in Fig. 3.7. A growing tendency for both horizons can be appreciated, being $N = 24$ considerably less affected than $N = 12$, however is important to take into consideration that $N = 24$ is less sensitive to changes than $N = 12$.

Once that we have asserted the robust behavior of Algorithm 1 against missing data, we proceed to implement the filter with real data, available from [5], containing multiple missing measurements. Fig. 3.8 sketches estimates of spectral features of CO concentration in the time span of 20–40 days. In this database, 3 discrete points (one-hour averaged) are lost at about 22nd hour and 24 points are lost in a span from about 29 to 30 hours. The lost data are indicated with $\alpha = 0$ in Algorithm 1. As can be seen, prediction organized in Algorithm 1 with lines 5–7 allows getting quite accurate estimates, while the basic algorithm (lines 5–7 removed) demonstrates extensive excursions, which values can be unacceptable.

3.5.2 Temperature Estimation with Missing Data

It is known that sensor accuracy is affected in many cases by the environment and interference from common industrial noise sources [108]. Therefore, averaging is often applied to improve accuracy. However, simple averaging that is typically invoked in such cases may

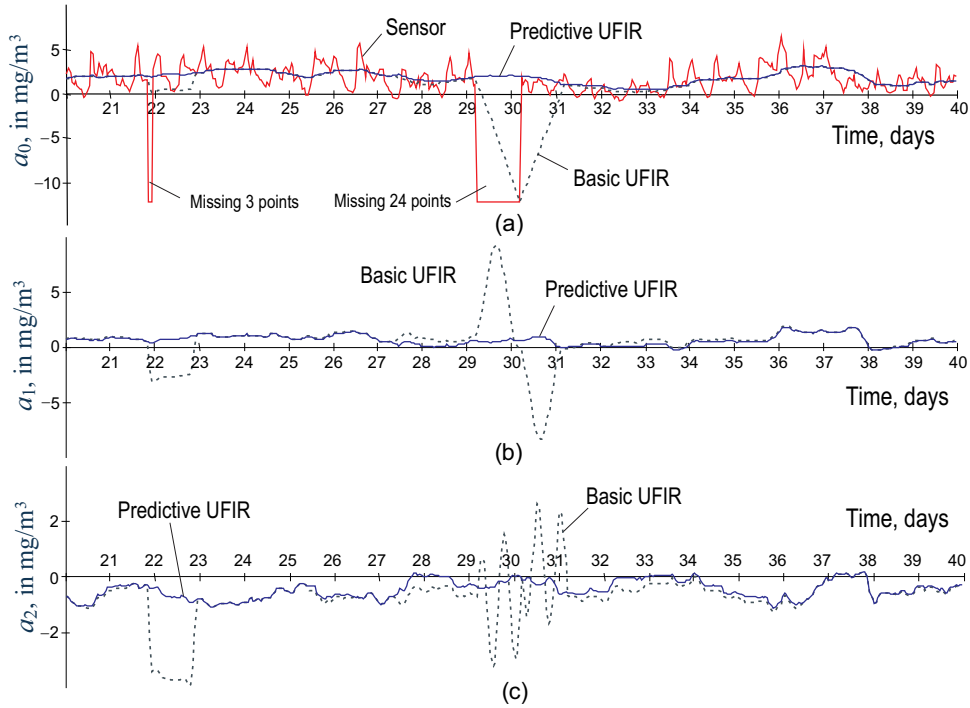


Figure 3.8: Spectral features of CO concentration with missing data for $N = 24$: (a) a_{0k} , (b) a_{1k} , and (c) a_{2k} . “Predictive UFIR” is by Algorithm 1 and “Basic UFIR” is by Algorithm 1 with lines 5–7 removed.

produce unacceptably large bias. The UFIR filter minimizes the bias errors.

In the second example, we consider measurements of temperature available from [5]. The archive contains missing data, which we set to zero. Referring to practical needs [4], we proceed with one-week averaging. To find UFIR estimates, we suppose that temperature changes linearly on an averaging horizon and specify the state-space model with $u_k = 0$, $e_k = 0$, $H_k = [1 \ 0]$, and

$$A_k = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix},$$

where $\tau = t_k - t_{k-1}$ is the sampling time. The noise components w_k and v_k are supposed to be zero mean. Other noise statistics and distributions are not known exactly and we ignore them. We finally specify matrix $C_{m,s}$ given by (3.8) as

$$C_{m,s} = \begin{bmatrix} 1 & -\tau \\ 1 & 0 \end{bmatrix}$$

and run Algorithm 1.

The estimation errors can be watched in Fig. 3.9 for $N = 24 \times 7 = 168$ that corresponds to a one-week horizon. Traditionally, we employ two options. The “Predictive UFIR” estimate is produced by Algorithm 1. To find the “Basic UFIR” estimate, we remove lines 5–7 from Algorithm 1 or suppose that $\alpha = 1$ over all measurements. As can be seen, the basic estimates are accompanied with large excursions in response to missing data, similarly to Fig. 3.8. On the contrary and in an agreement with Fig. 3.8, no excursions are observed at the output of the predictive UFIR filter. To emphasize the difference between the basic and predictive estimates, in Fig. 3.9b we show a five-week zoomed part of Fig. 3.9a. This figure suggests that the bridge made by Algorithm 1 over the uncertainty gap can be very efficient.

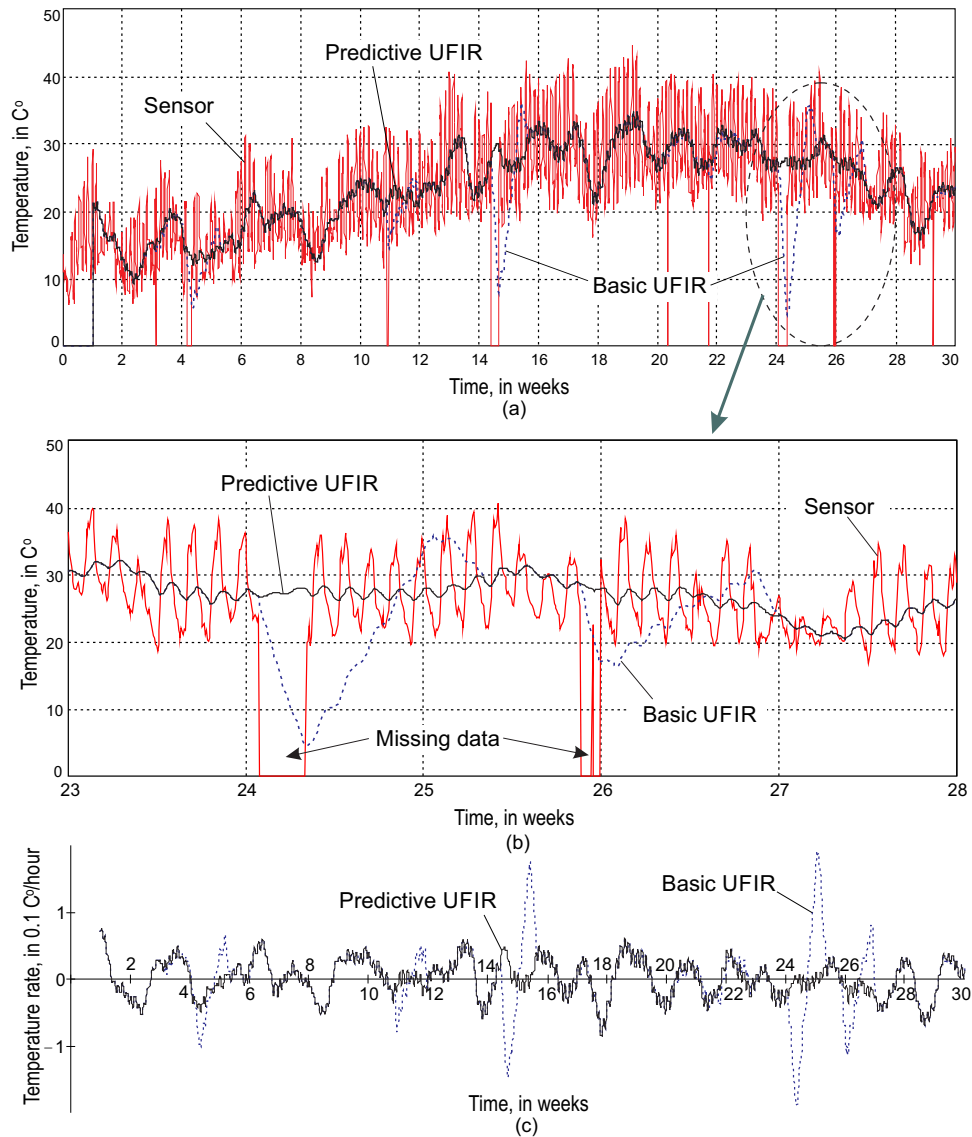


Figure 3.9: Estimates of temperature on a one-week horizon, $N = 168$, with missing data: (a) temperature measurements and estimates, (b) zoomed part of measurements and estimates, and (c) temperature rate estimates. “Predictive UFIR” is by Algorithm 1 and “Basic UFIR” is by Algorithm 1 with lines 5–7 removed or $\alpha = 1$.

Chapter 4

Design of UFIR Filter with Consensus on Measurements for WSNs

In the previous chapter we introduced a fast iterative UFIR filter with prediction capabilities, suitable to be implemented in a smart sensor. In this chapter we further our research by developing an average consensus estimator with consensus on measurements. The algorithm takes advantages of the unbiased finite impulse response filter technology to achieve superior performance against the popular Kalman filter under noise uncertainties.

4.1 Introduction

In industrial applications, wireless sensor networks (WSNs) are used to provide environmental sensing, condition monitoring, and process automation [109, 110]. Each node in the WSN interacts with a few neighbors, estimates the desired quantity \mathcal{Q} (temperature, velocity, pressure, etc.), and cooperatively passes data through other nodes to a central station [111]. Because \mathcal{Q} is typically measured over a big number of nodes in the presence of noise, optimal estimators are often used [17, 16, 11]. To make it possible to estimate \mathcal{Q} in real time with low computational complexity, distributed filtering has been introduced [22, 23] based on consensus on measurements [26, 112], on estimates [41], and on informa-

tion [29]. For a scalar quantity $Q(t)$, the consensus-based approach has been developed in [113, 26, 41, 114, 25]. For vector deterministic quantity $\vec{Q}(t)$, the consensus was found in [23] assuming noisy links in *Ad Hoc* WSNs, in [115] for the estimation and smoothing of vector random quantities, and in [116, 117] for random links. Kalman filtering is most widely used in design of consensus-based estimators [16, 11, 118]. However, *optimality* of the Kalman filter (KF) does not always go along with *robustness*, *scalability*, and *fault tolerance* required by the WSNs [42, 43]. Therefore, great efforts were made to improve the KF performance [16, 45, 44]. But what practice suggests is that robustification of the KF typically does not lead to essential progress [119, 34]. Better robustness demonstrate filters operating with finite data [120, 47].

Methods of data processing over finite data have attracted attention of designers of WSNs in recent years. In [48], a moving average estimator was designed for weak observability. A consensus finite-horizon H_∞ approach was developed in [3] under missing measurements and extended in [49, 50] to time-varying nonlinear networks. In [94], a two-stage recursive structure with norm-bounded parameter uncertainty was designed for the robust KF to operate on finite horizons similarly to the finite memory approach [34].

Beyond these solutions, FIR filtering offers several other fast algorithms which may efficiently be used in WSNs. A receding horizon (RH) Kalman FIR filter designed in [33] operates similarly to KF on finite horizons. For deterministic time-invariant control systems, a fast recursion-based algorithm was developed in [51]. An iterative p -shift unbiased FIR (UFIR) algorithm proposed in [43] completely ignores the noise statistics and initial values while reducing the output noise variance as a reciprocal of the horizon length. The p -shift UFIR estimator provides filtering with $p = 0$, RH filtering with $p = 1$, $|p|$ -lag smoothing with $p < 0$, and p -step prediction with $p > 0$ [53]. An important feature of the UFIR estimate is an ability to becomes practically optimal on large horizons. Besides, the performance of the UFIR filter can be improved by adapting the generalized noise power gain (GNPG) to operation conditions [54]. Fast optimal FIR algorithms were also designed [47, 55] and some other developments on FIR filtering can be found in [56, 57, 58, 59]. Hence, methods of FIR filtering can be used in design of robust WSNs.

Despite the scientific research on the subject, solutions still have not been addressed

to practitioners. For this reason, an objective of this research is to design a distributed UFIR filter based on supporting data-fusion average consensus on measurements and to show that it is more robust than the distributed KF. Because we introduce a new technique for WSNs, we do it only for a single quantity $\mathcal{Q}(t)$. In order to stress a link between the distributed UFIR filter and the distributed KF, we mostly save notations adopted in [26].

4.2 Model and Problem Formulation

We consider a manufacturing environment with quantity $\mathcal{Q}(t)$ of interest, whose dynamics is represented with a K -state vector $x_k \in \mathbb{R}^K$. We assume that the environment is covered with a WSN consisting of n nodes. Each i th, $i \in [1, n]$, node provides linear measurements of $\mathcal{Q}(t)$ as $y_k^{(i)} = H_k^{(i)}x_k + v_k^{(i)} \in \mathbb{R}^p$, where $H_k^{(i)} \in \mathbb{R}^{p \times K}$, $p \leq K$ is the number of the measured states, and $v_k^{(i)}$ is the measurement noise. In discrete time index k , the state-space model is

$$x_k = A_k x_{k-1} + B_k w_k, \quad (4.1)$$

$$y_k = H_k x_k + v_k, \quad (4.2)$$

where $y_k = [y_k^{(1)T} \dots y_k^{(n)T}]^T \in \mathbb{R}^{np}$ is the measurement vector, $A_k \in \mathbb{R}^{K \times K}$, $H_k = [H_k^{(1)T} \dots H_k^{(n)T}]^T \in \mathbb{R}^{np \times K}$, and B_k has proper dimensions. The zero mean mutually uncorrelated white Gaussian noise vectors, w_k and $v_k = [v_k^{(1)T} \dots v_k^{(n)T}]^T \in \mathbb{R}^{np}$, have the covariances Q_k and $R_k = \text{diag}[R_k^{(1)T} \dots R_k^{(n)T}]^T \in \mathbb{R}^{np \times np}$, respectively. We assign $\hat{x}_{k|r}$ to be an estimate of x_k at k via measurements up to and including at time-index r . We also employ the following variables: $\hat{x}_k^- \triangleq \hat{x}_{k|k-1}$ is the prior state estimate, $P_k^- = E\{(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T\}$ is the prior estimate covariance, \hat{x}_k is the posterior state estimate, and $P_k = E\{(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\}$ is the posterior error covariance.

In centralized WSNs, measurements pass to a central station, where the estimation of

\mathcal{Q} can be provided using the centralized KF (cKF) [26],

$$P_k^- = A_k P_{k-1} A_k^T + B_k Q_k B_k^T, \quad (4.3)$$

$$P_k = [(P_k^-)^{-1} + H_k^T R_k^{-1} H_k]^{-1}, \quad (4.4)$$

$$K_k = P_k H_k^T R_k^{-1}, \quad (4.5)$$

$$\hat{x}_k^- = A_k \hat{x}_{k-1}, \quad (4.6)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - H_k \hat{x}_k^-). \quad (4.7)$$

Provided P_0 , x_0 , Q_k , and R_k , the cKF optimally estimates state \hat{x}_k of $\mathcal{Q}(t_k)$ that has two disadvantages. Communication must be organized between the central station and each of the nodes. Real-time estimation may be an issue in view of the computational complexity when H_k acquires large dimensions.

To reduce the computational complexity, the cKF estimate (4.7) can be represented using (4.5) as

$$\begin{aligned} \hat{x}_k &= \hat{x}_k^- + K_k (y_k - H_k \hat{x}_k^-) \\ &= \hat{x}_k^- + P_k (H_k^T R_k^{-1} y_k - H_k^T R_k^{-1} H_k \hat{x}_k^-). \end{aligned} \quad (4.8)$$

By introducing two aggregate quantities: a vector $z_k \in \mathbb{R}^K$ of fused average-consensus sensor data and a fused average-consensus inverse covariance matrix $S_k \in \mathbb{R}^{K \times K}$,

$$z_k = \frac{1}{n} \sum_{i=1}^n z_k^{(i)} = \frac{1}{n} \sum_{i=1}^n H_k^{(i)T} R_k^{(i)-1} y_k^{(i)}, \quad (4.9)$$

$$S_k = \frac{1}{n} H_k^T R_k^{-1} H_k = \frac{1}{n} \sum_{i=1}^n H_k^{(i)T} R_k^{(i)-1} H_k^{(i)}, \quad (4.10)$$

and obtaining z_k by a low-pass consensus filter and S_k by a band-pass consensus filter [26], the micro-KF (μ KF) can be formulated as [26]

$$P_{\mu k}^- = A_k P_{\mu(k-1)} A_k^T + B_k Q_{\mu k} B_k^T, \quad (4.11)$$

$$P_{\mu k} = [(P_{\mu k}^-)^{-1} + S_k]^{-1}, \quad (4.12)$$

$$\hat{x}_k^- = A_k \hat{x}_{k-1}, \quad (4.13)$$

$$\hat{x}_k = \hat{x}_k^- + P_{\mu k} (z_k - S_k \hat{x}_k^-), \quad (4.14)$$

where $P_{\mu k} = nP_k$ is the μ KF gain, which is also the μ KF error covariance, and $Q_{\mu k} = nQ_k$. The μ KF operates n -times faster than the centralized Kalman filter (cKF) while producing equal estimates by “ideal” consensus.

The problem now formulates as follows. Given a WSN represented with (4.1) and (4.2), we would like to design a micro-UFIR (μ UFIR) filter for average consensus on measurements and show that the μ UFIR filter is more robust than the μ KF against modeling errors in not well specified noise environments. We also wish to investigate the trade-off between the μ KF and μ UFIR algorithms under harsh operation conditions based on the WSN simulation.

4.3 Distributed UFIR Filtering

Many issues of poor performance of the KF in WSNs are due to insufficient information about the initial values, x_0 and P_0 , and noise statistics, Q_k and R_k [121]. The iterative UFIR filter [43, 97] operates with N data points on a horizon $[m, k]$, from $m = k - N + 1$ to k , and ignores x_0 , P_0 , Q_k , and R_k at some (practically insignificant) expense in accuracy. The only tuning parameter required by the UFIR filter to minimize the mean squared error (MSE) is the optimal averaging horizon of N_{opt} points [43].

The batch UFIR estimate is given by [43]

$$\hat{x}_k = \bar{K}_{m,k} Y_{m,k} \quad (4.15a)$$

$$= (C_{m,k}^T C_{m,k})^{-1} C_{m,k}^T Y_{m,k}, \quad (4.15b)$$

where the extended observation vector $Y_{m,k}$ and mapping matrix $C_{m,k}$ are represented as

$$Y_{m,k} = [y_m^T \ y_{m+1}^T \ \cdots \ y_k^T]^T, \quad (4.16)$$

$$C_{m,k} = \begin{bmatrix} H_m(\mathcal{F}_k^{m+1})^{-1} \\ H_{m+1}(\mathcal{F}_k^{m+2})^{-1} \\ \vdots \\ H_{k-1}A_k^{-1} \\ H_k \end{bmatrix}, \quad (4.17)$$

and the product of system matrices is defined by

$$\mathcal{F}_k^r = \begin{cases} A_k A_{k-1} \dots A_r, & r < k+1 \\ I & r = k+1 \\ 0 & r > k+1 \end{cases}. \quad (4.18)$$

The UFIR filter gain $\bar{K}_{m,k}$ is given by the product

$$\bar{K}_{m,k} = G_k C_{m,k}^T, \quad (4.19)$$

in which the GNPG G_k is computed by [52]

$$G_k = \bar{K}_{m,k} \bar{K}_{m,k}^T = (C_{m,k}^T C_{m,k})^{-1}. \quad (4.20)$$

Below, we provide fast forms of (4.15b) for centralized UFIR (cUFIR) and micro UFIR (μ UFIR) filtering in WSNs.

4.3.1 Centralized UFIR Filter

In the Kalman-like UFIR filtering [43], the estimate at k is obtained iteratively using an auxiliary variable l beginning with $l = m + K$ and ending when $l = k$. The initial estimate at $l = m + K - 1$ is obtained using (4.15b) in a short batch form on a horizon $[m, m + K - 1]$. Provided G_{m+K-1} by (4.20) and \hat{x}_{m+K-1} by (4.15b), the estimate \hat{x}_k is computed iteratively as listed in Algorithm 3. The optimal horizon N_{opt} for the cUFIR filter can be found at the test stage by minimizing the trace of the estimation error P_k as

$$N_{\text{opt}} = \arg \min_N \{\text{tr } P_k(N)\} \quad (4.21)$$

or by utilizing observations with no reference as shown in [43]. Note that iterations will require about N_{opt} times more computation time than for the cKF.

The MSE matrix P_k can be found for the cUFIR filter as in [97]. Given the initial value P_{l-1} , the MSE matrix P_l can be updated with l changing from $k - N + K + 1$ to k as

$$P_l^- = A_l P_{l-1} A_l^T + B_l Q_l B_l^T, \quad (4.22)$$

$$P_l = (I - K_l H_l) P_l^- (I - K_l H_l)^T + K_l R_l K_l^T, \quad (4.23)$$

to take the true value when $l = k$.

Algorithm 3: Centralized UFIR Filtering Algorithm**Data:** y_k, N **Result:** \hat{x}_k

```

1 begin
2   for  $k = N - 1 : \infty$  do
3      $m = k - N + 1, \quad s = m + K - 1;$ 
4      $G_s = (C_{m,s}^T C_{m,s})^{-1};$ 
5      $\tilde{x}_s = G_s C_{m,s}^T Y_{m,s};$ 
6     for  $l = s + 1 : k$  do
7        $G_l = [H_l^T H_l + (A_l G_{l-1} A_l^T)^{-1}]^{-1};$ 
8        $\tilde{x}_l = A_l \tilde{x}_{l-1} + G_l H_l^T (y_l - H_l A_l \tilde{x}_{l-1});$ 
9     end for
10     $\hat{x}_k = \tilde{x}_k;$ 
11  end for
12 end

```

4.3.2 Micro-UFIR Filter

The consensus approach can now be applied to Algorithm 3. That can be done if to write the cUFIR estimate as

$$\begin{aligned}
\hat{x}_l &= \hat{x}_l^- + K_l (y_l - H_l \hat{x}_l^-) \\
&= \hat{x}_l^- + G_l (H_l^T y_l - H_l^T H_l \hat{x}_l^-) \\
&= \hat{x}_l^- + G_{\mu l} (s_l - L_l \hat{x}_l^-),
\end{aligned} \tag{4.24}$$

where $G_{\mu l} = nG_l$ is the *micro-UFIR filter GNPG* and

$$s_l = \frac{1}{n} \sum_{i=1}^n s_l^{(i)} = \frac{1}{n} \sum_{i=1}^n H_l^{(i)T} y_l^{(i)}, \tag{4.25}$$

$$L_l = \frac{1}{n} H_l^T H_l = \frac{1}{n} \sum_{i=1}^n H_l^{(i)T} H_l^{(i)}. \tag{4.26}$$

In contrast to (4.10), matrix (4.26) does not involve the inverse measurement noise covariance. Therefore, the μ UFIR filter will require only one consensus filter.

The *micro-UFIR filter* (μ UFIR) can then be designed similarly to the μ KF as

$$G_{\mu l} = [L_l + (A_l G_{\mu(l-1)} A_l^T)^{-1}]^{-1}, \quad (4.27)$$

$$\hat{x}_l^- = A_l \hat{x}_{l-1}, \quad (4.28)$$

$$\hat{x}_l = \hat{x}_l^- + G_{\mu l} (s_l - L_l \hat{x}_l^-), \quad (4.29)$$

where the bias correction gain $G_{\mu l}$ is also the GNPG.

Fast computation of the initial values $G_{\mu(l-1)}$ and \hat{x}_{l-1} can be provided using the filtered consensus values of L_k and s_k as in the following.

Transform the inverse of G_s using (4.18) to

$$\begin{aligned} G_s^{-1} &= C_{m,s}^T C_{m,s} \\ &= n \sum_{j=0}^{K-1} (\mathcal{F}_s^{m+1+j})^{-T} L_{m+j} (\mathcal{F}_s^{m+1+j})^{-1} \end{aligned} \quad (4.30)$$

that yields

$$G_{\mu s} = n G_s = \left[\sum_{j=0}^{K-1} (\mathcal{F}_s^{m+1+j})^{-T} L_{m+j} (\mathcal{F}_s^{m+1+j})^{-1} \right]^{-1}. \quad (4.31)$$

Also, transform the initial state \hat{x}_s as

$$\begin{aligned} \tilde{x}_s &= G_s C_{m,s}^T Y_{m,s} = \frac{1}{n} G_{\mu s} C_{m,s}^T Y_{m,s} \\ &= G_{\mu s} \sum_{j=0}^{K-1} (\mathcal{F}_s^{m+1+j})^{-T} s_{m+j}. \end{aligned} \quad (4.32)$$

The number K of the process states is typically small in WSNs that makes the sums short in (4.31) and (4.32). In particular, if a quantity \mathcal{Q} is represented with two states, $K = 2$, then the initial values can be computed as

$$G_{\mu s} = (A_{m+1}^{-T} L_m A_{m+1}^{-1} + L_{m+1})^{-1}, \quad (4.33)$$

$$\tilde{x}_s = G_{\mu s} (A_{m+1}^{-T} s_m + s_{m+1}). \quad (4.34)$$

What can now be observed is that the μ UFIR filter is a blind given N_{opt} and robust alternative to the μ KF as it does not require any information about the processes. A pseudo code of the μ UFIR algorithm is given as Algorithm 4. To specify the MSE matrix

Algorithm 4: Micro-UFIR Filtering Algorithm**Data:** s_k, L_k, N **Result:** \hat{x}_k

```

1 begin
2   for  $k = N - 1 : \infty$  do
3      $m = k - N + 1, \quad s = m + K - 1;$ 
4      $G_{\mu s} = \left[ \sum_{j=0}^{K-1} (\mathcal{F}_s^{m+1+j})^{-T} L_{m+j} (\mathcal{F}_s^{m+1+j})^{-1} \right]^{-1};$ 
5      $\tilde{x}_s = G_{\mu s} \sum_{j=0}^{K-1} \mathcal{F}_s^{m+1+j-T} s_{m+j};$ 
6     for  $l = s + 1 : k$  do
7        $G_{\mu l} = [L_l + (A_l G_{\mu(l-1)} A_l^T)^{-1}]^{-1};$ 
8        $\tilde{x}_l = A_l \tilde{x}_{l-1} + G_{\mu l} (s_l - L_l A_l \tilde{x}_{l-1});$ 
9     end for
10     $\hat{x}_k = \tilde{x}_k;$ 
11  end for
12 end

```

$P_{\mu k}$, transform $\varepsilon_l = x_l - \hat{x}_l$ using (4.1) and (4.29) to

$$\begin{aligned}
\varepsilon_l &= A_l \varepsilon_{l-1} + B_l w_l - G_{\mu l} s_l + G_{\mu l} L_l A_l \hat{x}_{l-1} \\
&= (I - G_{\mu l} L_l) (A_l \varepsilon_{l-1} + B_l w_l) \\
&\quad + G_{\mu l} \frac{1}{n} \sum_{i=1}^n H_l^{(i)} v_l^{(i)}
\end{aligned} \tag{4.35}$$

and write $P_l = E\{\varepsilon_l \varepsilon_l^T\}$ as

$$\begin{aligned}
P_l &= (I - G_{\mu l} L_l) (A_l P_{l-1} A_l^T + B_l Q_l B_l^T) (I - G_{\mu l} L_l)^T \\
&\quad + \frac{1}{n} G_{\mu l} S_l G_{\mu l}^T,
\end{aligned} \tag{4.36}$$

where S_l is given by (4.10). Now involve $P_{\mu l} = n P_l$ and $Q_{\mu l} = n Q_l$ from the μ KF and arrive at

$$\begin{aligned}
P_{\mu l} &= (I - G_{\mu l} L_l) (A_l P_{\mu(l-1)} A_l^T + B_l Q_{\mu l} B_l^T) \\
&\quad \times (I - G_{\mu l} L_l)^T + G_{\mu l} S_l G_{\mu l}^T.
\end{aligned} \tag{4.37}$$

By changing l from $k - N + K + 1$ to k , matrix $P_{\mu l}$ can now be updated iteratively at k .

4.4 Robustness of micro-Filters

In this section, we are going to show that the μ UFIR filter has better robustness than the μ KF.

4.4.1 Imprecisely Defined Noise Statistics

Exact knowledge about the noise covariances $Q_{\mu k}$ and $R_{\mu k}$ is typically unavailable in WSNs. We model errors in $Q_{\mu k}$ and $R_{\mu k}$ by the correction coefficients p and q as $Q_{\mu k} \leftarrow p^2 Q_{\mu k}$ and $R_k \leftarrow q^2 R_k$. The μ UFIR filter does not require the noise statistics and is thus protected against such errors. To ascertain effect of $\{p, q\} \neq 1$ on μ KF, we consider a stationary mode implying $P_{\mu k}^- \cong P_{\mu(k)} \cong P_{\mu(k-1)}$ and transform (4.12) to the discrete-time algebraic Lyapunov equation [101]

$$P_{\mu k}^- - A_k P_{\mu k}^- A_k^T = p^2 B_k Q_{\mu k} B_k^T, \quad (4.38)$$

which solution is known to be an infinite sum [102]

$$P_{\mu k}^- = p^2 \sum_{v=0}^{\infty} A_k^v B_k Q_{\mu k} B_k^T (A_k^v)^T = p^2 \Sigma_k. \quad (4.39)$$

By substituting $S_k \leftarrow \frac{1}{q^2} S_k$, the bias correction gain (4.12) can be rewritten as

$$P_{\mu k} = \left(\frac{1}{p^2} \Sigma_k^{-1} + \frac{1}{q^2} S_k \right)^{-1} \quad (4.40)$$

and, by $z_k \leftarrow \frac{1}{q^2} z_k$, we have

$$\begin{aligned} \hat{x}_k &= \hat{x}_k^- + P_{\mu k} (z_k - S_k \hat{x}_k^-) \\ &= \hat{x}_k^- + \left(\frac{\Sigma_k^{-1}}{p^2} + \frac{S_k}{q^2} \right)^{-1} \left(\frac{z_k}{q^2} - \frac{S_k}{q^2} \hat{x}_k^- \right) \\ &= \hat{x}_k^- + \bar{P}_{\mu k}(p, q) (z_k - S_k \hat{x}_k^-), \end{aligned} \quad (4.41)$$

where the estimation error covariance (bias correction gain) affected by p and q is

$$\bar{P}_{\mu k}(p, q) = \left(\frac{q^2}{p^2} \Sigma_k^{-1} + S_k \right)^{-1}. \quad (4.42)$$

Relation (4.42) suggests that the optimal value $\bar{P}_{\mu k} = P_{\mu k}$ is achieved when $p = q = 1$. Otherwise, random errors will dominate if $\bar{P}_{\mu k} > P_{\mu k}$ and the bias errors will grow if $\bar{P}_{\mu k} < P_{\mu k}$. One thus may conclude that the μ KF is less robust than the μ UFIR filter against errors in the noise covariances. The μ KF will retain supremacy when both p and q are near unity. Otherwise, the μ UFIR filter may produce more accuracy.

4.4.2 Temporary Model Errors

Any estimator produces extra errors when the WSN undergoes unpredictable impacts. We model such errors by scaling the system matrix during a short time as $A_k \leftarrow \alpha A_k$. We also allow $Q_{\mu k} \leftarrow p^2 Q_{\mu k}$ and $R_{\mu k} \leftarrow q^2 R_{\mu k}$. For the μ KF and μ UFIR filter, we thus have, respectively,

$$P_{\mu k} = \left[(\alpha^2 A_k P_{\mu(k-1)} A_k^T + p^2 B_k Q_{\mu k} B_k^T)^{-1} + \frac{1}{q^2} S_k \right]^{-1}, \quad (4.43)$$

$$G_{\mu l} = [(\alpha^2 A_l G_{\mu(l-1)} A_l^T)^{-1} + L_l]^{-1}, \quad (4.44)$$

and observe that the less robust μ KF will produce more bias errors if $p < 1$ and/or $q > 1$.

4.4.3 Temporary Measurement Errors

To model short-time measurement errors, we substitute H_k with βH_k , where $\beta = 1$ corresponds to accurate measurements. By letting $S_k \leftarrow \beta^2 S_k$, $z_k \leftarrow \beta z_k$, $L_k \leftarrow \beta^2 L_k$, and $s_k \leftarrow \beta s_k$, we transform the μ KF and μ UFIR estimates to

$$\begin{aligned} \hat{x}_k &= \hat{x}_k^- + \beta [(A_k P_{\mu(k-1)} A_k^T + B_k Q_{\mu k} B_k^T)^{-1} \\ &\quad + \beta^2 S_k]^{-1} (z_k - \beta S_k \hat{x}_k^-), \end{aligned} \quad (4.45)$$

$$\begin{aligned} \hat{x}_l &= \hat{x}_l^- + \beta [(A_l G_{\mu(l-1)} A_l^T)^{-1} + \beta^2 L_l]^{-1} \\ &\quad \times (s_l - \beta L_l \hat{x}_l^-). \end{aligned} \quad (4.46)$$

As can be seen, effect of $\beta \neq 1$ can be gained in the μ KF by errors in the noise statistics and the μ KF is thus less robust than μ UFIR to node errors.

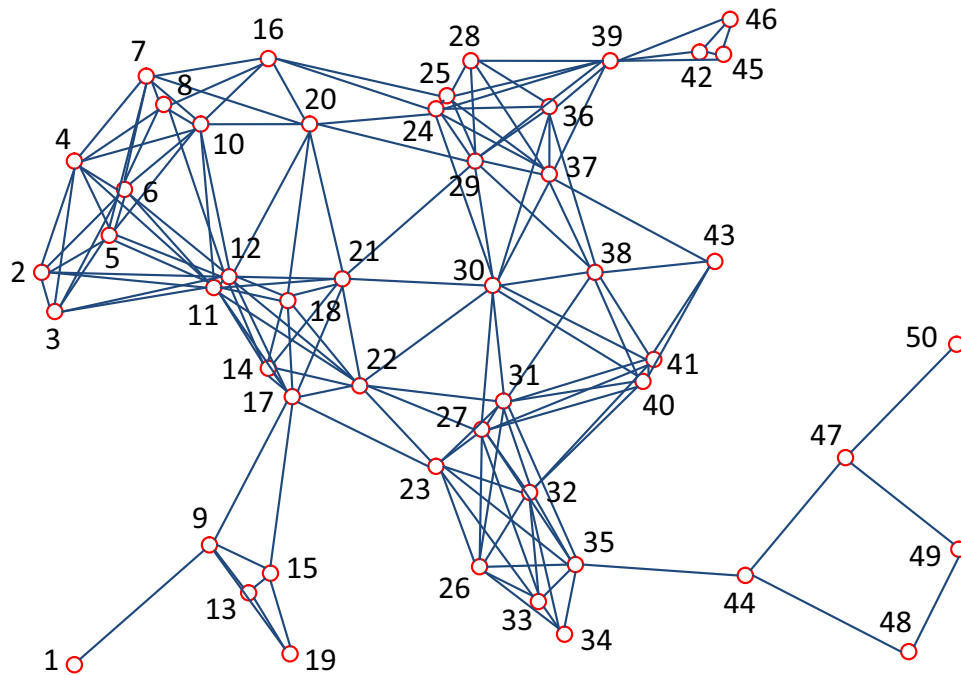


Figure 4.1: Simulated WSN with 50 nodes randomly placed in coordinates of $-40 \text{ m} \leq x \leq 40 \text{ m}$ and $-30 \text{ m} \leq y \leq 30 \text{ m}$: 145 links are due to the node range of 14 m.

4.5 Example of Applications

In this section, we compare errors produced by nodes with the cKF, cUFIR filter, μ KF and μ UFIR filter under diverse operation conditions. A WSN is organized to cover a territory of $-40 \text{ m} \leq x \leq 40 \text{ m}$ and $-30 \text{ m} \leq y \leq 30 \text{ m}$. The nodes are placed randomly with coordinates uniformly distributed along axes x and y . The Laplace matrix of the WSN graph has 50×50 dimensions and the network is shown in Fig. 4.1. Each node communicates with neighbors within a range of 14 m. Table 4.1 sorts the nodes by the number of links.

As can be observed, the maximum number of the links is 13 (node 12) and most frequently repeated numbers are 3 (six nodes), 6 (seven nodes), 7 (seven nodes), 8 (six nodes), 9 (eight nodes), and 10 (five nodes). It is supposed that data can be transferred cooperatively to a central station.

We consider a vehicle traveling circularly on a ground space limited with coordinates covered by the WSN. The two-state vector x_k unites the coordinate x as the first state

Table 4.1: Nodes Sorted by the Number of Links with Neighbors

Number of links	Nodes
1	1,50
2	48,49
3	13,19,42,44,45,46,47
4	15,34,43
5	28
6	52,3,8,9,16,25,33
7	5,7,14,26,36,40,41
8	4,20,23,24,35,38
9	6,10,17,18,21,32,37,39
10	11,22,27,29,31
11	30
12	–
13	12

x_{1k} and the coordinate y as the second state x_{2k} . The vehicle dynamics (4.1) is governed starting with $x_0 = [-10 \text{ m } 10 \text{ m}]^T$ by the time-invariant system matrix

$$A = \begin{bmatrix} a & -b \\ b & a \end{bmatrix},$$

where $a = 0.999$ and $b = 0.04$, and B identity. The noise components in vector $w_k = [w_{1k} \ w_{2k}]^T$ have the variance $\sigma_w^2 = 0.01 \text{ m}^2$ and the covariance is $Q = \text{diag}[\sigma_w^2 \ \sigma_w^2]$. Practice suggests that Q is typically not known exactly and vehicle dynamics may undergo temporary unpredictable changes caused by some external force. We take it into account by letting p^2Q and substituting a in A with $a + \delta_k$, where δ_k is an amendment to be specified latter. It is assumed that each node is equipped to measure the vehicle coordinate x (first state x_{1k}) with $H^{(i)} = [1 \ 0]$ and is linked with n neighbors as specified in (4.2). Each noise component $v_k^{(i)}$ has the variance $\sigma_v^2 = 25 \text{ m}^2$ and thus $R^{(i)} = [\sigma_v^2]$. Because noise statistics are not available for each individual sensor and a set of sensors is typically specified in

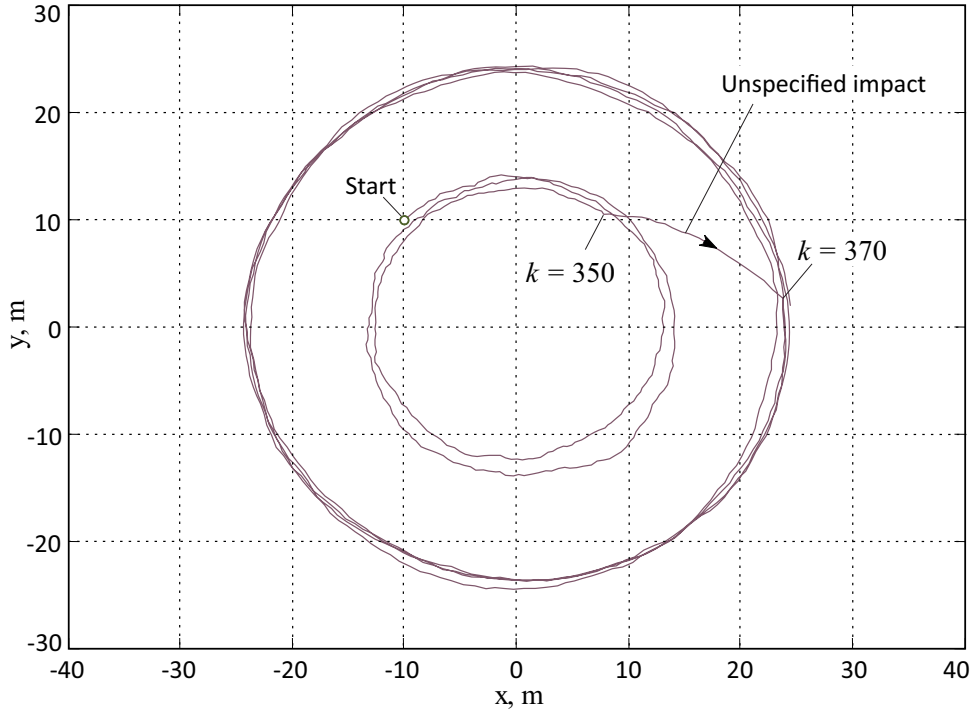


Figure 4.2: Vehicle traveling circularly on a ground space starting with $x_0 = -10\text{ m}$ and $y_0 = 10\text{ m}$ at $k = 0$. Unpredictably, an external force affects the movement from $k = 350$ to $k = 370$.

the worst case, we scale $R^{(i)}$ as $q^2 R^{(i)}$, where q represents an error. At the test stage, the optimal horizon was found to be $N_{\text{opt}} = 88$.

A simulated trajectory of a moving vehicle is shown in Fig. 4.2 for $\delta_k = 0.03$ when $350 \leq k \leq 370$ and $\delta_k = 0$ otherwise. As can be seen, an external force changes the trajectory in a time span of $350 \leq k \leq 370$ that subsequently results in a larger radius.

4.5.1 Tracking Under Incomplete Information About Noise

In the first scenario, we suppose that the noise covariances are not known exactly, set $p = 0.5$ and $q = 2$, substitute $Q_k \leftarrow p^2 Q_k$, $Q_{\mu k} \leftarrow p^2 Q_{\mu k}$, $R_k \leftarrow q^2 R_k$, and $R_{\mu k} \leftarrow q^2 R_{\mu k}$, and investigate abilities of the filters to track the vehicle trajectory shown in Fig. 4.2. Note that p and q may take much larger values if no information is available about noise. Estimation errors produced by the nodes with different numbers of the links (Table 4.1) are shown in Fig. 4.3 that leads to several important conclusions:

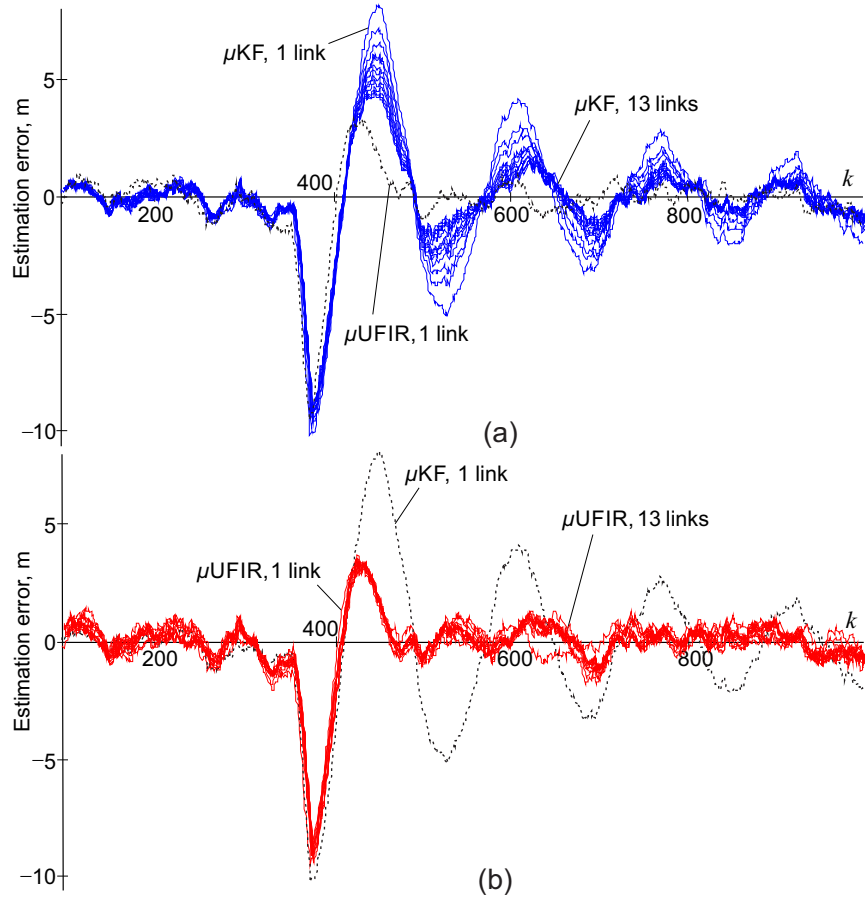


Figure 4.3: Estimation errors produced by nodes with micro-filters along the coordinate x : (a) μ KF with the μ UFIR filter (1 link) as a benchmark and (b) μ UFIR filter with the μ KF (1 link) as a benchmark.

- When no disturbance affects the trajectory ($0 \leq k < 350$), both the μ KF and μ UFIR filter produce similar estimates, which are almost invariant to the number of the links.
- The μ KF responds to the disturbance $\delta_k = 0.03$, $350 \leq k \leq 370$, with larger excursions and oscillating transients, while the transient in the μ UFIR filter is finished at $k = 458$, after $N_{\text{opt}} = 88$ points. Thus, the μ UFIR filter is more robust.
- An increase in the number of the links improves the μ KF performance but does not affect essentially the μ UFIR filter. Even with one link, the μ UFIR filter produces more accuracy than the μ KF with 13 links.

The root MSEs (RMSEs) sketched in Fig. 4.4 give a more precise picture. For the

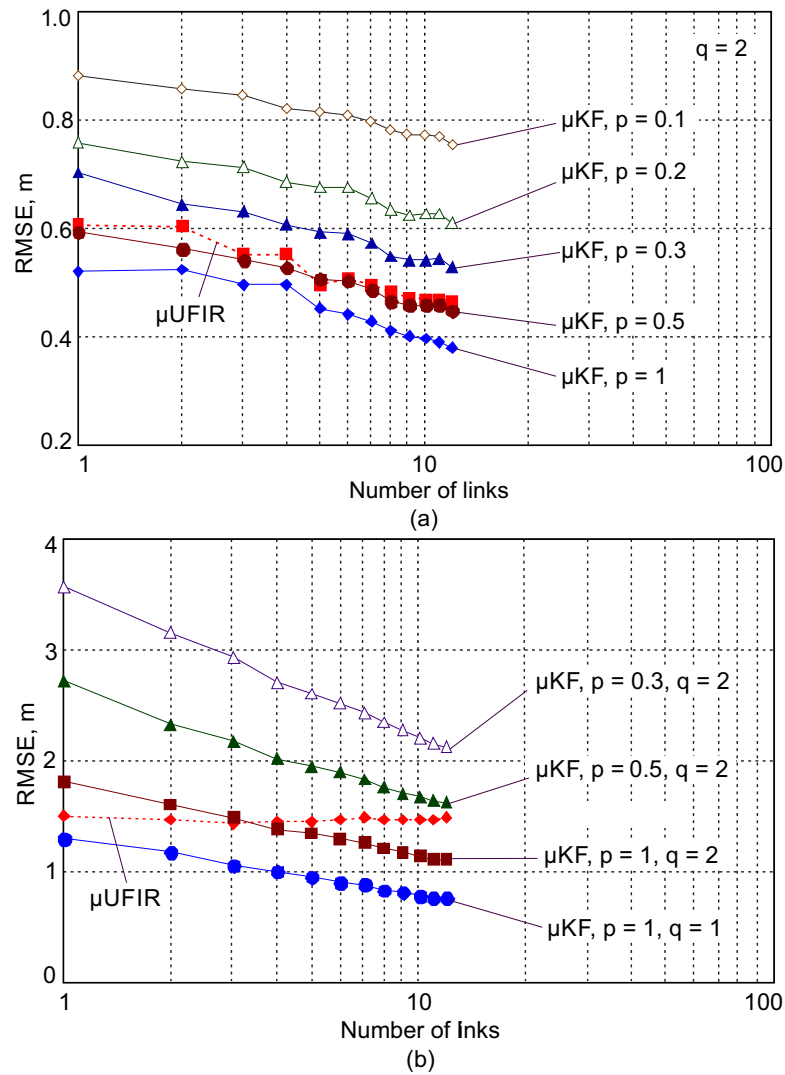


Figure 4.4: RMSEs produced by the nodes with micro filters as functions of the number of the links along the coordinate x : (a) $\delta_k = 0$ and (b) $\delta_k \neq 0$.

undisturbed movement, $\delta_k = 0$, both filters increase an accuracy at about 12 cm as the number of the links grows from 1 to 13 (Fig. 4.4a). In the disturbed case (Fig. 4.4b), the μ UFIR filter becomes invariant to the number of the links while the μ KF keeps reducing errors as the number of the links grows. Again we notice that, in both cases, the μ UFIR filter demonstrates better robustness, because the μ KF performance is highly affected by p and q .

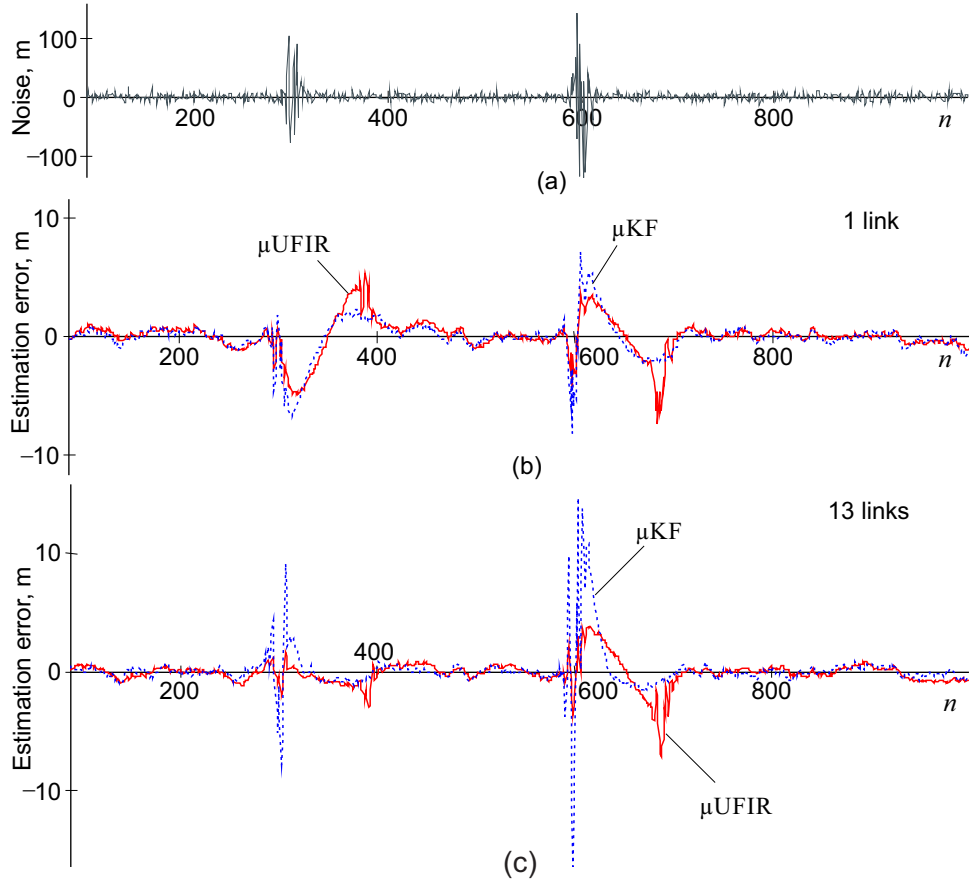


Figure 4.5: Effect of the impulsive measurement noise on estimation errors: (a) impulsive noise, (b) 1-link node, and (c) 13-links node.

4.5.2 Effect of Unspecified Impulsive Noise

In the second scenario, a vehicle travels with no external impact, $\delta_k = 0$, but the WSN is attacked by impulsive noise caused, for example, by stamping. We simulate this noise as a Gaussian process, whose variance experiences an increase at two points, $k = 300$ and $k = 600$, as shown in Fig. 4.5a. We set $R_k \leftarrow q^2 R_k$ and $R_{\mu k} \leftarrow q^2 R_{\mu k}$ with $q = 2$ and $p = 2$. The estimation errors are sketched in Fig. 4.5b and Fig. 4.5c. The results reveal that both filters respond to the noise splashes with excursions and transients such that the produced errors become almost equal for one link (Fig. 4.5b). However, an increase of the number of the links makes the μ KF less accurate at $k = 300$ and $k = 600$ (Fig. 4.5c).

One may arrive at similar conclusions by observing the RMSEs sketched for $q = 1$ and diverse values of p in Fig. 4.6. As can be seen, the μ UFIR filter is more accurate here

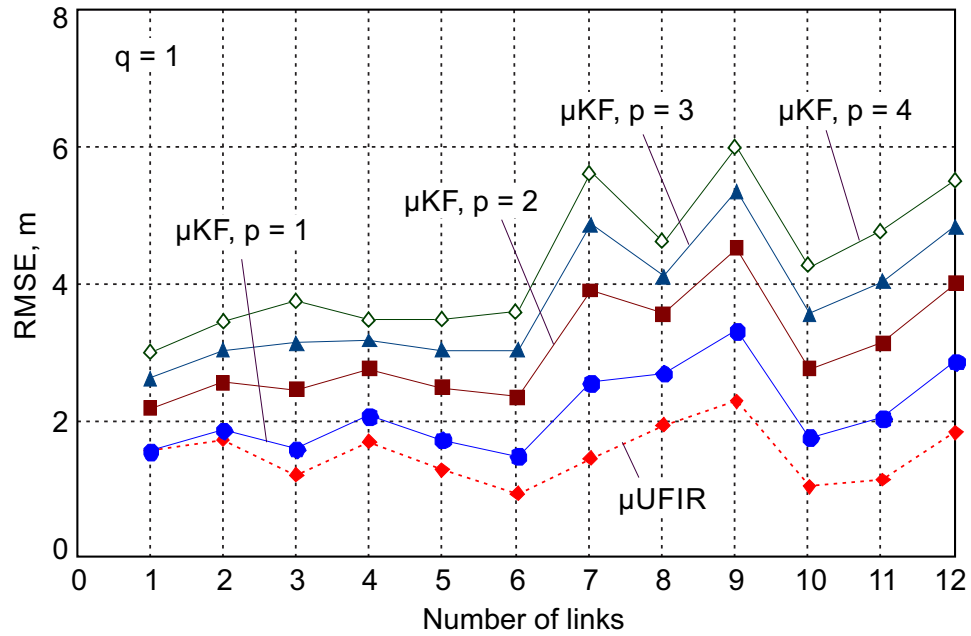


Figure 4.6: RMSEs produced by nodes with the μKF and μUFIR filter under the unspecified impulsive measurement noise with $q = 1$ and $p > 1$.

for any number of the links. On the contrary, an increase in the number of the links and growing p result in larger errors produced by the μKF .

Chapter 5

Design of UFIR Filter With Consensus on Estimates for WSNs

In the previous chapter we developed a UFIR filter with consensus on measurements where a low-pass and a band-pass filter were implemented on the measurements and transition matrices before the μ UFIR filter. The pre-filtering process reduces considerably the execution time of the algorithm, however the results are exactly the same as a centralized UFIR filter so, in essence, the consensus on estimates strategy implemented on the UFIR filter will only reduce its implementation time. For this reason, in this chapter, we propose a UFIR filter with consensus on estimates that compensates a centralized filter with an optimal factor and a consensus protocol, obtaining better performance in terms of robustness and lower root mean squared error.

5.1 Introduction

In recent decades, wireless sensor networks (WSNs) have gone through intensive developments due to remarkable technological advances and cost reduction achieved in smart sensing. Accordingly, many ubiquitous and large scale solutions were proposed for WSN applications in industry, healthcare, and services [12, 13, 14]. Generally, large scale nodes deployment allows organizing WSNs with a redundant number of measurements of a desired quantity Q . Further processing of data corrupted by noise makes it possible to reduce

the estimation error reciprocally to the number of the nodes. This rule allows exploiting a massive number of low-cost sensors with a proper balance between the desirable accuracy and cost. The best noise reduction is typically achieved in WSNs using optimal estimation and fusion techniques [15, 11, 16, 17], which remain in the developments for diverse WSN structures [18, 19, 20, 21].

Although WSNs can be organized to be either centralized, decentralized, or distributed [11], the latter manifested themselves as most powerful and flexible [22, 23, 24]. In such structures, nodes do not transmit information over long distances that improves battery life. Also, distributed filtering can be organized using different kinds of consensus in the WSN that increases estimation accuracy. Consensus can be found between the measurements [26], estimates [1], information matrices [40], and other dynamic features of the WSN [122, 123]. It also improves the network fault tolerance by providing an access to estimates of more than one node. This especially matters for WSNs involved in monitoring vast environmental areas, such as crop fields and forests.

An idea of consensus on measurements was originally formulated by Olfati-Saber in [26] and then developed by other authors [112, 21]. The algorithms employ the Kalman filter (KF) as an estimation platform to process data with known error covariances. The flaws are that 1) all sensors are supposed to have the same model and 2) the KF-based schemes demonstrate an ability to diverge under errors in the noise statistics. The consensus on estimates was also originally proposed by Olfati-Saber [1] and then developed in [118, 124, 125] using the KF. Although higher accuracy was reported, the approach did not take advantage of useful information, which can be extracted from the noise statistics. To improve the convergence, a consensus on information was later considered by Li [40]. It was however noticed that the accuracy may degrade due to an adopted fusion rule [40].

The KF is most popular in the design of fusion algorithms [30, 31] owing to simplicity, optimality, and low computational burden. For example, the KF-based structure developed in [1] requires each node to locally aggregate an estimate and the covariance matrix together with those provided by neighbors and then find a consensus on estimates [118, 124]. In [125], an extended KF was used to estimate a position of a static target under specific operation conditions. Due to a lack of prior knowledge of the noise statistics, the pedestrian position

was first roughly estimated in [126] by an artificial neural network and then improved using the KF.

Real world dictates that WSNs and state estimators must be robust, scalable, energy efficient, and fault tolerant [11]. The KF does always meet these needs in view of often incomplete information of the process and environment [42, 43]. Better robustness is inherent to finite impulse response (FIR) structures [46, 47] and H_∞ filters [9]. For WSNs, a moving average FIR estimator was designed in [48] assuming a weak observability, a consensus finite-horizon H_∞ filter developed in [3] for missing data, and an unbiased FIR (UFIR) filter proposed for consensus on measurements in [21] and for measurements with delayed and missing data in [127]. In [128, 129], a solution was found for WSNs using a linear optimal weights-based fusion technique assuming correlated noise. First attempts to design fast UFIR algorithms for WSNs with consensus on estimates were made in [130], where the design was provided only for autonomous WSNs and requires information about noise. The necessity to have a more general and robust solution motivates our present work.

In this chapter, we design a distributed UFIR (dUFIR) filtering algorithm for WSNs with consensus on estimates and show its better performance against the distributed KF (dKF) and distributed H_∞ (d H_∞) filter. We admit that nodes in industrial applications do not necessarily measure the same states and can be tuned either individually or to some specifications. We also notice that fast, optimal, and low-memory algorithms are of a top priority for smart sensors and follow this line.

5.2 Model and Problem Formulation

Let us view a WSN as an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where each vertex $v^{(i)} \in \mathcal{V}$ is a node and each link is an edge of set \mathcal{E} , for $i \in \mathcal{I} = \{1, \dots, n\}$ and $n = |\mathcal{V}|$. As stated in [25], nodes $v^{(i)}$ and $v^{(j)}$ reach an agreement if and only if the states are related as $x^{(i)} = x^{(j)}$, $\{i, j\} \in \mathcal{I}, i \neq j$. If so, the WSN reaches a consensus with a common value called the *group decision value*. Because a perfect consensus is unavailable due to process noise, a consensus protocol is required to minimize a total disagreement in the WSN. It is provided

by minimizing the Laplacian potential of graph $\Psi = \frac{1}{2}x^T Lx$, where L is the Laplacian matrix. A linear distributed protocol for minimizing the total disagreement is formulated as

$$u^{(i)} = \sum_j^J [x^{(j)} - x^{(i)}].$$

In a general scenario of distributed WSNs, we will suppose that dynamics of a quantity \mathcal{Q} is represented with K states and controlled at each discrete time index k . We will also assume that $J \triangleq J_k$ nodes measure \mathcal{Q} at each k and data are available during finite time due to limited resources. Accordingly, we represent \mathcal{Q} with linear state and observation equations as

$$x_k = F_k x_{k-1} + E_k u_k + B_k w_k, \quad (5.1)$$

$$y_k = H_k x_k + v_k, \quad (5.2)$$

$$y_k^{(i)} = H_k^{(i)} x_k + v_k^{(i)}, \quad (5.3)$$

where $x_k \in \mathbb{R}^K$, $u_k \in \mathbb{R}^M$, $F_k \in \mathbb{R}^{K \times K}$, $E_k \in \mathbb{R}^{K \times M}$, and $B_k \in \mathbb{R}^{K \times L}$. The i th, $i \in [1, J]$, node measures x_k by $y_k^{(i)} \in \mathbb{R}^p$, $p \leq K$, with $H_k^{(i)} \in \mathbb{R}^{p \times K}$ and each node has J inclusive neighbors. Local data $y_k^{(i)}$ are united in the observation vector $y_k = [y_k^{(1)T} \dots y_k^{(J)T}]^T \in \mathbb{R}^{Jp}$ with $H_k = [H_k^{(1)T} \dots H_k^{(J)T}]^T \in \mathbb{R}^{Jp \times K}$. Noise vectors $w_k \in \mathbb{R}^L$ and $v_k = [v_k^{(1)T} \dots v_k^{(J)T}]^T \in \mathbb{R}^{Jp}$ are zero mean, not obligatorily white Gaussian, uncorrelated, and with the covariances $Q_k = E\{w_k w_k^T\} \in \mathbb{R}^{L \times L}$, $R_k = \text{diag}[R_k^{(1)T} \dots R_k^{(J)T}]^T \in \mathbb{R}^{Jp \times Jp}$, and $R_k^{(i)} = E\{v_k^{(i)} v_k^{(i)T}\}$.

To apply FIR filtering, we extend (5.1)–(5.3) on a horizon $[m, k]$ of N points, from $m = k - N + 1$ to k as [10]

$$X_{m,k} = A_{m,k} x_m + S_{m,k} U_{m,k} + D_{m,k} W_{m,k}, \quad (5.4)$$

$$Y_{m,k} = C_{m,k} x_m + L_{m,k} U_{m,k} + M_{m,k} W_{m,k} + V_{m,k}, \quad (5.5)$$

where $X_{m,k} = [x_m^T \ x_{m+1}^T \ \dots \ x_k^T]^T$, $Y_{m,k} = [y_m^T \ y_{m+1}^T \ \dots \ y_k^T]^T$, $U_{m,k} = [u_m^T \ u_{m+1}^T \ \dots \ u_k^T]^T$, $W_{m,k} = [w_m^T \ w_{m+1}^T \ \dots \ w_k^T]^T$, and $V_{m,k} = [v_m^T \ v_{m+1}^T \ \dots \ v_k^T]^T$. The extended matrices are

$$A_{m,k} = [I \ F_{m+1}^T \ \dots \ (\mathcal{F}_k^{m+1})^T]^T, \quad (5.6)$$

$$S_{m,k} = \begin{bmatrix} E_m & 0 & \dots & 0 & 0 \\ F_{m+1}E_m & E_{m+1} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathcal{F}_{k-1}^{m+1}E_m & \mathcal{F}_{k-1}^{m+2}E_{m+1} & \dots & E_{k-1} & 0 \\ \mathcal{F}_k^{m+1}E_m & \mathcal{F}_k^{m+2}E_{m+1} & \dots & F_k E_{k-1} & E_k \end{bmatrix}, \quad (5.7)$$

$C_{m,k} = \bar{C}_{m,k}A_{m,k}$, $L_{m,k} = \bar{C}_{m,k}S_{m,k}$, and $M_{m,k} = \bar{C}_{m,k}D_{m,k}$, where

$$\bar{C}_{m,k} = \text{diag}(H_m \ H_{m+1} \ \dots \ H_k), \quad (5.8)$$

$$\mathcal{F}_r^g = \begin{cases} F_r F_{r-1} \dots F_g, & g < r+1 \\ I, & g = r+1 \\ 0, & g > r+1 \end{cases}, \quad (5.9)$$

and $D_{m,k}$ is defined by substituting E_k with B_k in (5.7). The local extended vector $Y_{m,k}^{(i)} = \left[y_m^{(i)T} \ y_{m+1}^{(i)T} \ \dots \ y_k^{(i)T} \right]^T$ is defined via (5.4), where $C_{m,k}$, $L_{m,k}$, $M_{m,k}$, and $V_{m,k}$ are substituted with $C_{m,k}^{(i)}$, $L_{m,k}^{(i)}$, $M_{m,k}^{(i)}$ and $V_{m,k}^{(i)}$.

The problem can now be formulated as follows. Given model (5.4)–(5.5), we would like to design a batch dUFIR filter with optimal consensus on estimates by minimizing the mean square error (MSE). We also wish to represent the batch dUFIR filter with a computationally efficient iterative form using recursions, test by simulations and real data, and compare to the dKF and d H_∞ filter.

5.3 dUFIR Filter with Consensus on Estimates

It follows from [10] that the distributed FIR estimate for model (5.4)–(5.5) with control signal $U_{m,k}$ can be written as

$$\hat{x}_k = \Theta_{m,k}Y_{m,k} + (\bar{S}_{m,k} - \Theta_{m,k}L_{m,k})U_{m,k}, \quad (5.10)$$

where

$$\bar{S}_{m,k} = [\mathcal{F}_k^{m+1}E_m \ \mathcal{F}_k^{m+2}E_{m+1} \ \dots \ F_k E_{k-1} \ E_k]$$

is the last row vector in $S_{m,k}$ and $\Theta_{m,k}$ is the FIR filter gain. The i th, $i \in [1, J]$, node provides a local estimate over $Y_{m,k}^{(i)}$ as $\hat{x}_k^{(i)}$. Then, referring to [1], the consensus between

local estimates can be found by introducing a vector

$$\Sigma_k = \sum_j^J [\hat{x}_k^{(j)} - \hat{x}_k^{(i)}],$$

combine it with (5.10), and write the estimate as

$$\hat{x}_k^c = \Theta_{m,k} Y_{m,k} + (\bar{S}_{m,k} - \Theta_{m,k} L_{m,k}) U_{m,k} + \lambda_k \Sigma_k, \quad (5.11)$$

where λ_k is a scaling factor to be optimized in the MSE sense.

Because we are concerned of dUFIR filtering, we claim that the filter obeys only the unbiasedness condition of

$$E\{\hat{x}_k^c\} = E\{\hat{x}_k^{(i)}\} = E\{x_k\} \quad (5.12)$$

to guarantee that the average estimate is equal to that of the model specified by the last N th row vector in (5.4) as

$$x_k = \mathcal{F}_k^{m+1} x_m + \bar{S}_{m,k} U_{m,k} + \bar{D}_{m,k} W_{m,k}, \quad (5.13)$$

where $\bar{D}_{m,k}$ is the last row vector in $D_{m,k}$.

By defining the error as $\varepsilon_k = x_k - \hat{x}_k^c$ and considering the error covariance $P_k = E\{\varepsilon_k \varepsilon_k^T\}$, the optimal factor λ_k^{opt} can further be found using (5.11) by minimizing the MSE as

$$\lambda_k^{\text{opt}} = \arg \min_{\lambda_k} \{\text{tr } P_k\}. \quad (5.14)$$

Referring to (5.10)–(5.14), the dUFIR filter can now be designed as in the following.

5.3.1 Batch dUFIR Filter

Given (5.4) and (5.5), the estimate (5.11) can be rewritten as

$$\begin{aligned} \hat{x}_k^c &= (I + J\lambda_k) [\Theta_{m,k} (Y_{m,k} - L_{m,k} U_{m,k}) + \bar{S}_{m,k} U_{m,k}] \\ &\quad - J\lambda_k [\Theta_{m,k}^{(i)} (Y_{m,k}^{(i)} - L_{m,k}^{(i)} U_{m,k}) + \bar{S}_{m,k} U_{m,k}], \end{aligned} \quad (5.15)$$

where gains $\Theta_{m,k}$ and $\Theta_{m,k}^{(i)}$ will obey the unbiasedness condition (5.12) if we follow [10] and find

$$\Theta_{m,k} = (I + J\lambda_k) G_k \mathcal{H}_{m,k}^T, \quad (5.16)$$

$$\Theta_{m,k}^{(i)} = G_k^{(i)} \mathcal{H}_{m,k}^{(i)T}, \quad (5.17)$$

where

$$\mathcal{H}_{m,k} = \begin{bmatrix} H_m(\mathcal{F}_k^{m+1})^{-1} \\ H_{m+1}(\mathcal{F}_k^{m+2})^{-1} \\ \vdots \\ H_{k-1}F_k^{-1} \\ H_k \end{bmatrix}, \quad (5.18)$$

$G_k = (\mathcal{H}_{m,k}^T \mathcal{H}_{m,k})^{-1}$ is the generalized noise power gain (GNPG) [52], $\mathcal{H}_{m,k}^{(i)}$ is defined by (5.18) by substituting H_l , $l \in [m, k]$, with $H_l^{(i)}$, and $G_k^{(i)} = (\mathcal{H}_{m,k}^{(i)T} \mathcal{H}_{m,k}^{(i)})^{-1}$ is the GNPG of the i th filter.

As can be seen, information required to compute $\Theta_{m,k}$ and $\Theta_{m,k}^{(i)T}$ is entirely provided by the K -state space model, which can be preloaded on the nodes. Thus, only measurement data will be sent by the nodes. This implies a reduction of the number of exchange messages and improves battery life.

5.3.2 Optimum Consensus Factor λ_k^{opt}

The optimum factor λ_k^{opt} can be found by solving the optimization problem (5.14). In doing so, we refer to (5.11) and (5.13), recall that noise is uncorrelated, and write the error covariance

$$P_k = E\{\varepsilon_k \varepsilon_k^T\} = E\{(x_k - \hat{x}_k^{(c)})(x_k - \hat{x}_k^{(c)})^T\}, \quad (5.19)$$

as function of λ_k and by solving the optimization problem (see Appendix A) we obtain the expression

$$\lambda_k^{\text{opt}} = -\frac{1}{J}(\tilde{\Theta}_{m,k} \bar{R}_{m,k} \tilde{\Theta}_{m,k}^T - G_k G_k^{(i)-1} \Theta_{m,k}^{(i)} \bar{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T}) \times \\ (\tilde{\Theta}_{m,k} \bar{R}_{m,k} \tilde{\Theta}_{m,k}^T - 2G_k G_k^{(i)-1} \Theta_{m,k}^{(i)} \bar{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T} + \Theta_{m,k}^{(i)} \bar{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T})^{-1}. \quad (5.20)$$

Let us now notice that the implementation of (5.15) with (5.20) may cause a computational problem due to large matrix dimensions and limited resources of smart sensors. Therefore, we will next derive computationally efficient and low-memory iterative forms for (5.15) and (5.20) using recursions and show that the computational cost can be greatly reduced.

5.3.3 Iterative dUFIR Algorithm Using Recursions

It follows that the batch dUFIR filter (5.15) with optimal factor λ_k^{opt} (5.20) requires only the data noise covariances, while the dKF also requires the initial values and system noise covariance. Since (5.15) and (5.20) are computationally inefficient due to large matrix dimensions with $N \gg 1$, recursions are required for both (5.15) and (5.20).

Recursions for dUFIR Filter

To compute (5.15) iteratively, we represent \hat{x}_k^c defined by (5.11) as a combination of the centralized estimate \hat{x}_k given by (5.10) and i th local estimate $\hat{x}_k^{(i)}$, which has the same structure as (5.10),

$$\hat{x}_k^c = (I + J\lambda_k)\hat{x}_k - J\lambda_k\hat{x}_k^{(i)}. \quad (5.21)$$

Iterative forms for \hat{x}_k and $\hat{x}_k^{(i)}$ can be found using recursions given in [10, 21]. Namely, for \hat{x}_k , one can employ

$$G_l = [H_l^T H_l + (F_l G_{l-1} F_l^T)^{-1}]^{-1}, \quad (5.22)$$

$$\hat{x}_l^- = F_l \hat{x}_{l-1} + E_l u_l, \quad (5.23)$$

$$\hat{x}_l = \hat{x}_l^- + G_l H_l^T (y_l - H_l \hat{x}_l^-), \quad (5.24)$$

where l is an iterative variable ranging from $s = k - N + K$ to k . Iterations (5.22)–(5.24) can be initialized with $G_{l-1} = G_s$ and $\hat{x}_{l-1} = \hat{x}_s$ in short batch forms of

$$G_s = (\mathcal{H}_{m,s}^T \mathcal{H}_{m,s})^{-1}, \quad (5.25)$$

$$\hat{x}_s = G_s \mathcal{H}_{m,s}^T (Y_{m,s} - L_{m,s} U_{m,s}) + \bar{S}_{m,s} U_{m,s}. \quad (5.26)$$

Reasoning similarly, $\hat{x}_k^{(i)}$ can be obtained via (5.22)–(5.26) by substituting each matrix (or vector) X with $X^{(i)}$.

Recursions for λ_k^{opt}

An iterative computation of factor λ_k^{opt} can be provided if to represent (5.20) as

$$\lambda_k^{\text{opt}} = -\frac{1}{J}(\alpha_k - G_k G_k^{(i)-1} \beta_k)(\alpha_k - 2G_k G_k^{(i)-1} \beta_k + \beta_k)^{-1}, \quad (5.27)$$

where $\alpha_k = \tilde{\Theta}_{m,k} \bar{R}_{m,k} \tilde{\Theta}_{m,k}^T$ and $\beta_k = \Theta_{m,k}^{(i)} \bar{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T}$, and use the recursions derived in Appendix B and Appendix C,

$$\alpha_k = G_k (H_k^T R_k H_k + F_k^{-T} G_{k-1}^{-1} \alpha_{k-1} G_{k-1}^{-1} F_k^{-1}) G_k \quad (5.28)$$

$$\beta_k = G_k^{(i)} (H_k^{(i)T} R_k^{(i)} H_k^{(i)} + F_k^{-T} G_{k-1}^{(i)-1} \beta_{k-1} G_{k-1}^{(i)-1} F_k^{-1}) G_k^{(i)}, \quad (5.29)$$

with initial values α_{k-1} and β_{k-1} computed in short batch forms as

$$\alpha_s = G_s \mathcal{H}_{m,s}^T \bar{R}_{m,s} \mathcal{H}_{m,s} G_s^T, \quad (5.30)$$

$$\beta_s = G_s^{(i)} \mathcal{H}_{m,s}^{(i)T} \bar{R}_{m,s}^{(i)} \mathcal{H}_{m,s}^{(i)} G_s^{(i)T}. \quad (5.31)$$

Iterative dUFIR Algorithm

A pseudo code of the designed iterative dUFIR algorithm with consensus on estimates is listed as Algorithm 5. Given N , Algorithm 5 starts computing the initial values at $s = m + K - 1$ and then updates the results beginning at $s + 1$ until the iterative variable l reaches k . It then computes λ_k^{opt} and finishes with the output estimate \hat{x}_k^c . Of importance is that for time-invariant systems factor λ_k^{opt} is constant and thus can be preloaded in a smart sensor to reduce the number of operations in Algorithm 5.

5.4 A Comparison of dUFIR and dKF Algorithms

To show a definitive advantage of the dUFIR filter, we bring along the dKF Algorithm 6 as designed in [1]. The following differences can be indicated:

- Factor λ_k^{opt} (5.20) of the dUFIR filter is affected by the data noise covariance R_k , but is invariant to system noise covariance Q_k , which is typically less known. In contrast, the dKF is affected by both these matrices.
- For equal sensors, $G_k = \frac{1}{J} G_k^{(i)}$ and $\mathcal{H}_{m,s}^T \bar{R}_{m,s} \mathcal{H}_{m,s} = J \mathcal{H}_{m,s}^{(i)T} \bar{R}_{m,s}^{(i)} \mathcal{H}_{m,s}^{(i)}$ make $\lambda_k^{\text{opt}} = 0$ and the dUFIR estimate becomes invariant to noise, while the dKF still depends on both R_k and Q_k .

Algorithm 5: Iterative dUFIR Filtering Algorithm**Data:** $y_k, R_k^{(i)}, R_k, N$ **Result:** \hat{x}_k 1 **begin**2 **for** $k = N - 1 : \infty$ **do**3 $m = k - N + 1, \quad s = m + K - 1;$ 4 $G_s = (\mathcal{H}_{m,s}^T \mathcal{H}_{m,s})^{-1};$ 5 $G_s^{(i)} = (\mathcal{H}_{m,s}^{(i)T} \mathcal{H}_{m,s}^{(i)})^{-1};$ 6 $\tilde{x}_s = G_s \mathcal{H}_{m,s}^T (Y_{m,s} - L_{m,s} U_{m,s}) + \bar{S}_{m,s} U_{m,s};$ 7 $\tilde{x}_s^{(i)} = G_s^{(i)} \mathcal{H}_{m,s}^{(i)T} (Y_{m,s}^{(i)} - L_{m,s}^{(i)} U_{m,s}) + \bar{S}_{m,s} U_{m,s};$ 8 $\alpha_s = G_s \mathcal{H}_{m,s}^T \bar{R}_{m,s} \mathcal{H}_{m,s} G_s^T;$ 9 $\beta_s = G_s^{(i)} \mathcal{H}_{m,s}^{(i)T} \bar{R}_{m,s}^{(i)} \mathcal{H}_{m,s}^{(i)} G_s^{(i)T};$ 10 **for** $l = s + 1 : k$ **do**11 $\hat{x}_l^- = F_l \hat{x}_{l-1} + E_l u_l;$ 12 $\hat{x}_l^{(i)-} = F_l \hat{x}_{l-1}^{(i)} + E_l u_l;$ 13 $G_l = [H_l^T H_l + (F_l G_{l-1} F_l^T)^{-1}]^{-1};$ 14 $G_l^{(i)} = [H_l^{(i)T} H_l^{(i)} + (F_l G_{l-1}^{(i)} F_l^T)^{-1}]^{-1};$ 15 $\hat{x}_l = \hat{x}_l^- + G_l H_l^T (y_l - H_l \hat{x}_l^-);$ 16 $\hat{x}_l^{(i)} = \hat{x}_l^{(i)-} + G_l^{(i)} H_l^{(i)T} (y_l^{(i)} - H_l^{(i)} \hat{x}_l^{(i)-});$ 17 $\alpha_l = G_l (H_l^T R_l H_l + F_l^{-T} G_{l-1}^{-1} \alpha_{l-1} G_{l-1}^{-1} F_l^{-1}) G_l;$ 18 $\beta_l = G_l^{(i)} (H_l^{(i)T} R_l^{(i)} H_l^{(i)} + F_l^{-T} G_{l-1}^{(i)-1} \beta_{l-1} G_{l-1}^{(i)-1} F_l^{-1}) G_l^{(i)};$ 19 **end for**20 $\lambda_k^{\text{opt}} = -\frac{1}{J} (\alpha_k - G_k G_k^{(i)-1} \beta_k) \times$ 21 $(\alpha_k - 2G_k G_k^{(i)-1} \beta_k + \beta_k)^{-1}; \hat{x}_k^c = (I + J \lambda_k) \tilde{x}_k - J \lambda_k \tilde{x}_k^{(i)};$ 22 **end for**23 **end**

Algorithm 6: Iterative dKF Algorithm

Data: $P_0^{(i)}, Q_k, R_k^{(j)}, \bar{x}_k^{(j)}, y_k^{(j)}, \bar{x}_0^{(i)} = x_0$

Result: $\hat{x}_k^{(i)}$

- 1 **begin**
- 2 **for** $k = 0 : \infty$ **do**
- 3 $z_k^{(j)} = H_k^{(j)T} R_k^{(j)-1} y_k^{(j)}, \quad \forall j \in J;$
- 4 $s_k^{(i)} = \sum_{j \in J} z_k^{(j)};$
- 5 $Z_k^{(j)} = H_k^{(j)T} R_k^{(j)-1} H_k^{(j)}, \quad \forall j \in J;$
- 6 $S_k^{(i)} = \sum_{j \in J} Z_k^{(j)} = H_k^T R_k^{-1} H_k;$
- 7 $M_k^{(i)} = (P_k^{(i)-1} + S_k^{(i)})^{-1};$
- 8 $\hat{x}_k^{(i)} = \bar{x}_k^{(i)} + M_k^{(i)}(s_k^{(i)} - S_k^{(i)} \bar{x}_k^{(i)}) + \epsilon M_k^{(i)} \sum_{j \in J} (\bar{x}_k^{(j)} - \bar{x}_k^{(i)});$
- 9 $P_k^{(i)} \leftarrow F_k M_k^{(i)} F_k^T + B_k Q_k B_k^T;$
- 10 $\bar{x}_k^{(i)} \leftarrow F_k \hat{x}_k^{(i)};$
- 11 **end for**
- 12 **end**

- For different sensors, one may assume that $H_k^{(i)} = H_k^{(j)}$ and $R_k^{(i)} = p^{(i)} R_k^*$, where R_k^* is an average covariance and $p^{(i)}$ is an error factor of the i th sensor, and write

$$\lambda_k^{\text{opt}} = -\frac{1}{J} \frac{\bar{P} - p^{(i)}}{\bar{P} + (J-2)p^{(i)}} I \quad (5.32)$$

where $P = [p^{(i)T} \dots p^{(j)T}]^T \in \mathbb{R}^J$ and \bar{P} is an average of P . It then follows that an increase in J diminishes λ_k^{opt} , as well as the effect of errors on λ_k^{opt} . We do not observe such an effect in the dKF.

Overall, it follows that the dKF is affected by both R_k and Q_k , while the dUFIR filter only by the difference between $R_k^{(i)}$ of individual sensors. When this difference approaches zero, the dUFIR filter becomes noise-invariant and thus more robust than the dKF.

Table 5.1: Nodes Sorted by the Number of Available Links with Neighbors

Number of links	Nodes
1	33
2	42
3	13,16,18,25,29,49
4	4,10,11,21,31,34,37,44
5	3,5,8,9,17,28,41
6	6,7,14,24,27,32,43,47,48
7	2,12,15,22,36
8	1,19,23,38,39,40,45
9	26,30,35,46,50
10	20

5.5 Examples of Applications

In this section, we will test the dUFIR Algorithm 5 along with the dKF algorithm described in [1]. A numerical example will be given for tracking of a circularly travelling and rapidly maneuvering object. Experimental verification will be provided for robot localization with known ground truth.

5.5.1 Maneuvering Object Tracking

It has been shown in [21] that the dUFIR filter is more robust than the dKF for consensus on measurements. We will now consider a similar example, but for consensus on estimates. A trajectory is simulated at 700 discrete points of an object moving circularly and maneuvering suddenly in a time span of $350 \leq k \leq 370$. The WSN is organized to cover an area of $-40 \text{ m} \leq (x, y) \leq 40 \text{ m}$ with 50 randomly placed nodes, each with a range of 14 m (Fig. 5.1). The links available for each node as listed in Table 5.1.

The model is specified with $F = \begin{bmatrix} a & b \\ -b & a \end{bmatrix}$, $a = 0.9996$, $b = 0.03$, and $B = I$. A part of the trajectory of $350 \leq k \leq 370$ is simulated by substituting a with $a + 0.03$. Gaussian

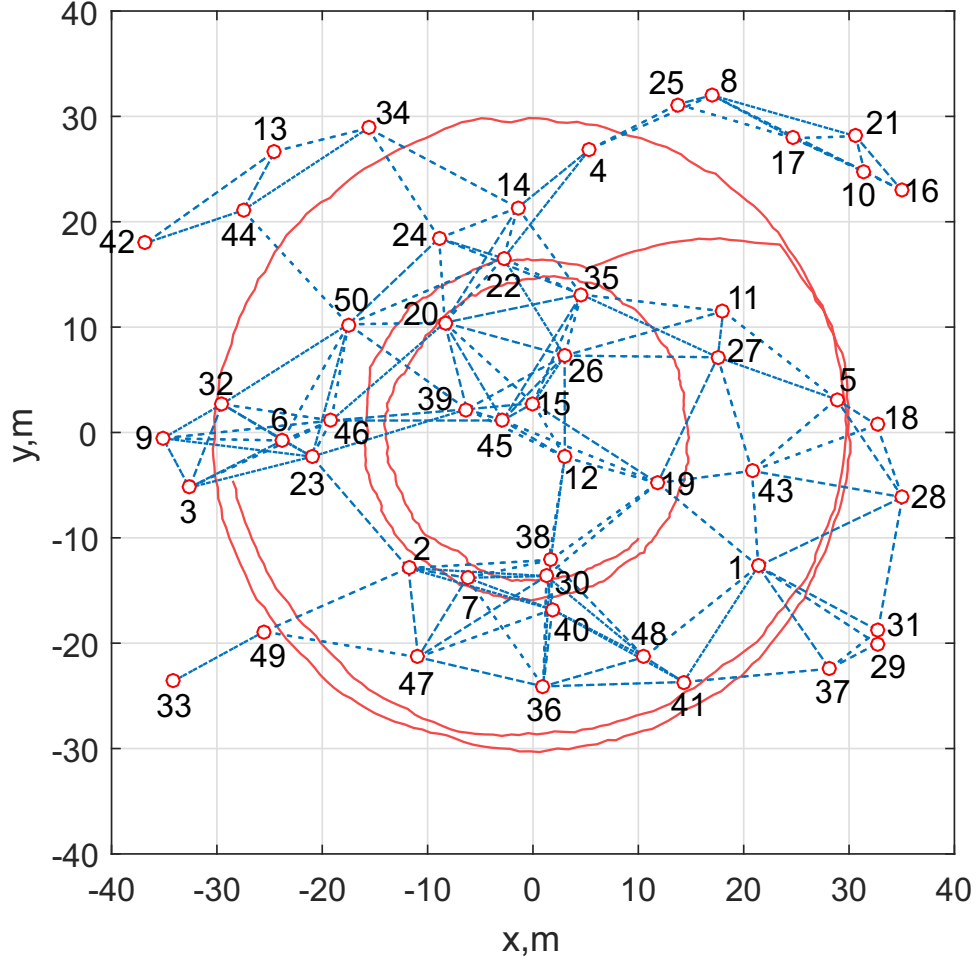


Figure 5.1: Simulated WSN with 50 nodes in the area of $-40 \text{ m} \leq (x, y) \leq 40 \text{ m}$ and 144 links (dashed) due to the nodes range of 14 m. The object trajectory (solid) is circular with a sudden maneuver changing the radius.

$w_k = [w_{1k} \ w_{2k}]^T$ is generated to have the standard deviation $\sigma_w = 0.1 \text{ m}$ and the covariance $Q = \text{diag}(\sigma_w^2 \ \sigma_w^2)$. Each sensor is assumed to measure either the first or second state and we thus set $H^{(i)} = [1 \ 0]$ for coordinate x and $H^{(i)} = [0 \ 1]$ for y. Gaussian v_k has the covariance $R_k^{(i)} = E\{v_k^{(i)} v_k^{(i)T}\} = (\sigma_v + \phi)^2$, where $\sigma_v = 5 \text{ m}$ and ϕ is uniformly distributed on $[-1, 1]$. An optimal horizon $N_{\text{opt}} = 88$ was found for the dUFIR filter at a test stage.

Effect of uncertainties in λ_k

To learn the effect of tuning factors α_k and β_k on λ_k^{opt} (5.27) via R_k and $R_k^{(i)}$, we allow $R_k^{(i)} \leftarrow q^2 R_k^{(i)}$ with q taking values from $\{1, 1.2, 1.4, 1.8, 2\}$. The results show that the

dUFIR output is affected negligibly by $1 \leq q \leq 2$ and an increase in the number of links decreases the dUFIR filter errors, as expected.

Errors in the noise covariances

To show that the dUFIR filter outperforms the dKF [1] under errors in the noise covariances, we allow $Q_k \leftarrow p^2 Q_k$ and $R_k^{(i)} \leftarrow q^2 R_k^{(i)}$ with $p = 0.5$ and $q = 2$. The H_∞ filter is generally more robust than the KF, but still not developed for consensus on estimates. To find out how the H_∞ filter [9] measures up to the dFIR filter, we modify the dKF algorithm to be the d H_∞ algorithm. We do it following [131] by substituting $M_k^{(i)}$ in Algorithm 6 with $M_k^{(i)} = (P_k^{(i)-1} + \theta I + S_k^{(i)})^{-1}$, where θ is a small positive scalar factor.

Comparative results sketched in Fig. 5.2 suggest that the dUFIR filter responds to uncertainties faster (during N points) than the dKF, which demonstrates lasting and oscillating transients with larger excursions. The d H_∞ filter outperforms the dUFIR filter with $\theta_1 = 0.80768$ valid for 10 links in a time span of $350 \leq k \leq 370$ and diverges beginning at $k = 650$. It is important that even an insignificant deviation from θ_1 to $\theta_2 = 0.80766$ turns the d H_∞ filter to divergence when an object starts maneuvering. Thus, the d H_∞ filter cannot be considered as a good candidate for real life WSN scenarios, at least in our case.

Figure 5.3 reveals that the dUFIR filter produces smaller errors with an increase in the number of the links that also speaks in favor of its higher robustness. From Fig. 5.4, one infers that large errors in the noise covariances make the dKF highly inaccurate, while the dUFIR filter is almost insensitive to such errors. Even the required R_k does not affect the dUFIR filter as much as the dKF.

5.5.2 UVG Localization over WSN for Measured Ground Truth

We next employ a trajectory of an unmanned ground vehicle (UVG) (robot) measured at 3000 points with a time step of $\tau = 0.01$ s and available for free use from the MagPIE dataset [6]. To localize the UVG, we simulate a WSN with 8 nodes deployed to cover the trajectory as shown in Fig. 5.5. A distance $\rho_k^{(i)}$ is measured by an i th time-of-flight (ToF) ranging sensor VL53L0X, which resolution is limited with $\Delta\rho_k^{(i)} = 4.8$ cm. An angular

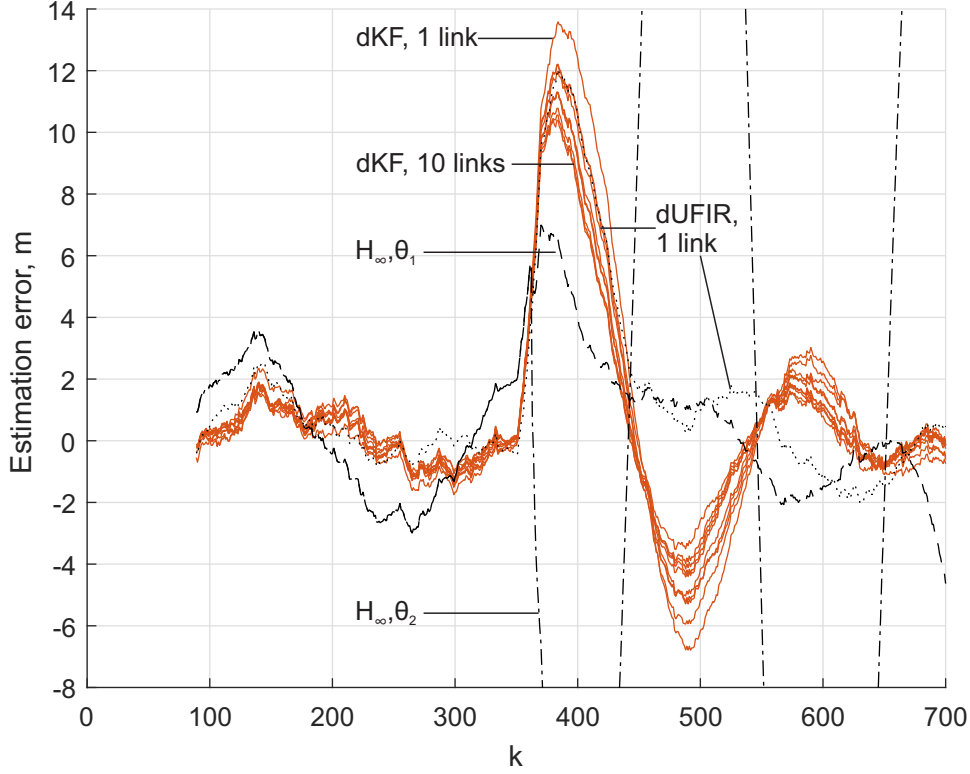


Figure 5.2: Estimation errors along axis x produced by dKF under errors in Q_k and $R_k^{(i)}$. The dUFIR filter errors for 1 link are shown as a benchmark. The dH_∞ estimates are shown for two tuning factors $\theta_1 = 0.80768$ and $\theta_2 = 0.80766$ with an extremely small difference.

position $\phi_k^{(i)}$ with respect to a basic line is provided by the i th vibration rejecting rate gyro data sheet ADXRS649 with a resolution of $\Delta\phi_k^{(i)} = 0.47^\circ$.

Each node has a range of 2 m and can communicate over 5 m. When the UVG travels within the node range, both $\rho_k^{(i)}$ and $\phi_k^{(i)}$ are measured, while the altitude is ignored. For data $\rho_k^{(i)} = \bar{\rho}_k^{(i)} + \Delta\rho_k^{(i)}$ and $\phi_k^{(i)} = \bar{\phi}_k^{(i)} + \Delta\phi_k^{(i)}$, where $\bar{\rho}_k^{(i)}$ and $\bar{\phi}_k^{(i)}$ are average values and $\Delta\rho_k^{(i)}$ and $\Delta\phi_k^{(i)}$ are white Gaussian and uncorrelated with the standard deviations $\sigma_{\rho k} = |\Delta\rho_k^{(i)}|/3$ and $\sigma_{\phi k} = |\Delta\phi_k^{(i)}|/3$, we represent the UVG Cartesian coordinates as

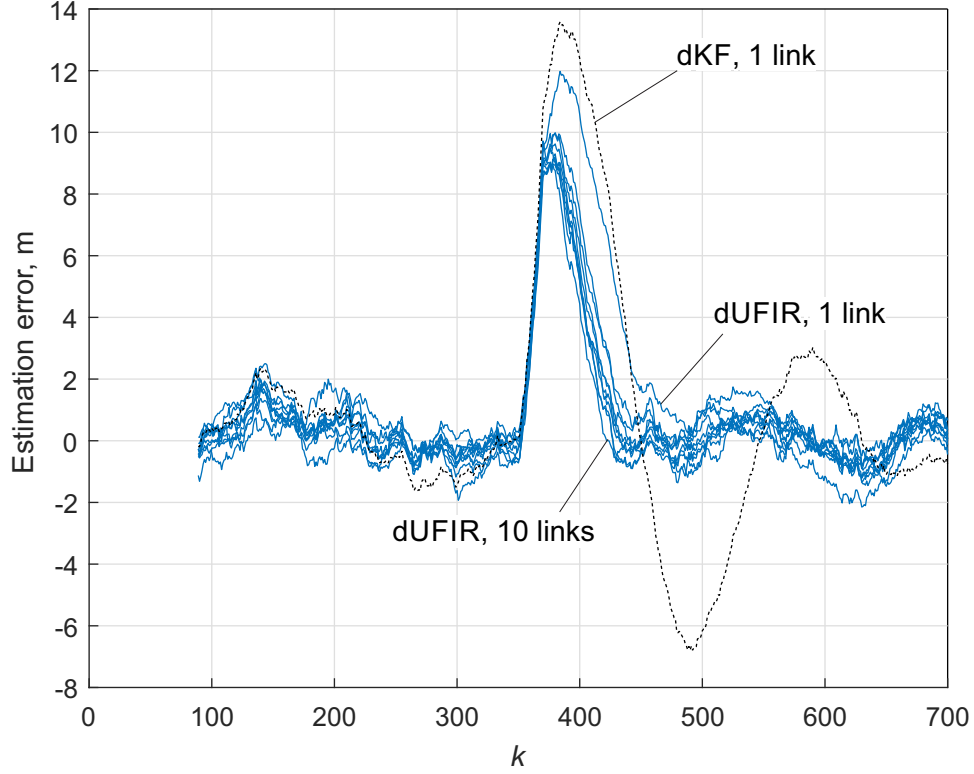


Figure 5.3: Estimation errors along axis x produced by the dUFIR filter under errors in Q_k and $R_k^{(i)}$. The dKF errors for 1 link are shown as a benchmark.

$x_k^{(i)} = \rho_k^{(i)} \cos \phi_k^{(i)} = \bar{x}_k^{(i)} + \Delta x_k^{(i)}$ and $y_k^{(i)} = \rho_k^{(i)} \sin \phi_k^{(i)} = \bar{y}_k^{(i)} + \Delta y_k^{(i)}$ and approximate with

$$\begin{aligned} x_k^{(i)} &= (\bar{\rho}_k^{(i)} + \Delta \rho_k^{(i)}) \cos(\bar{\phi}_k^{(i)} + \Delta \phi_k^{(i)}), \\ &\cong \bar{\rho}_k^{(i)} \cos \bar{\phi}_k^{(i)} + \Delta \rho_k^{(i)} \cos \bar{\phi}_k^{(i)} \\ &\quad - \Delta \phi_k^{(i)} \bar{\rho}_k^{(i)} \sin \bar{\phi}_k^{(i)}, \end{aligned} \quad (5.33)$$

$$\begin{aligned} y_k^{(i)} &= (\bar{\rho}_k^{(i)} + \Delta \rho_k^{(i)}) \sin(\bar{\phi}_k^{(i)} + \Delta \phi_k^{(i)}), \\ &\cong \bar{\rho}_k^{(i)} \sin \bar{\phi}_k^{(i)} + \Delta \rho_k^{(i)} \sin \bar{\phi}_k^{(i)} \\ &\quad + \Delta \phi_k^{(i)} \bar{\rho}_k^{(i)} \cos \bar{\phi}_k^{(i)}, \end{aligned} \quad (5.34)$$

where $\bar{x}_k^{(i)} = \bar{\rho}_k^{(i)} \cos \bar{\phi}_k^{(i)}$, $\bar{y}_k^{(i)} = \bar{\rho}_k^{(i)} \sin \bar{\phi}_k^{(i)}$, $\Delta x_k^{(i)} = \Delta \rho_k^{(i)} \cos \bar{\phi}_k^{(i)} - \Delta \phi_k^{(i)} \bar{\rho}_k^{(i)} \sin \bar{\phi}_k^{(i)}$, and $\Delta y_k^{(i)} = \Delta \rho_k^{(i)} \sin \bar{\phi}_k^{(i)} + \Delta \phi_k^{(i)} \bar{\rho}_k^{(i)} \cos \bar{\phi}_k^{(i)}$.

We next define the measurement noise variances as $\sigma_{xk}^{(i)2} = E\{\Delta x_k^{(i)2}\}$ and $\sigma_{yk}^{(i)2} =$

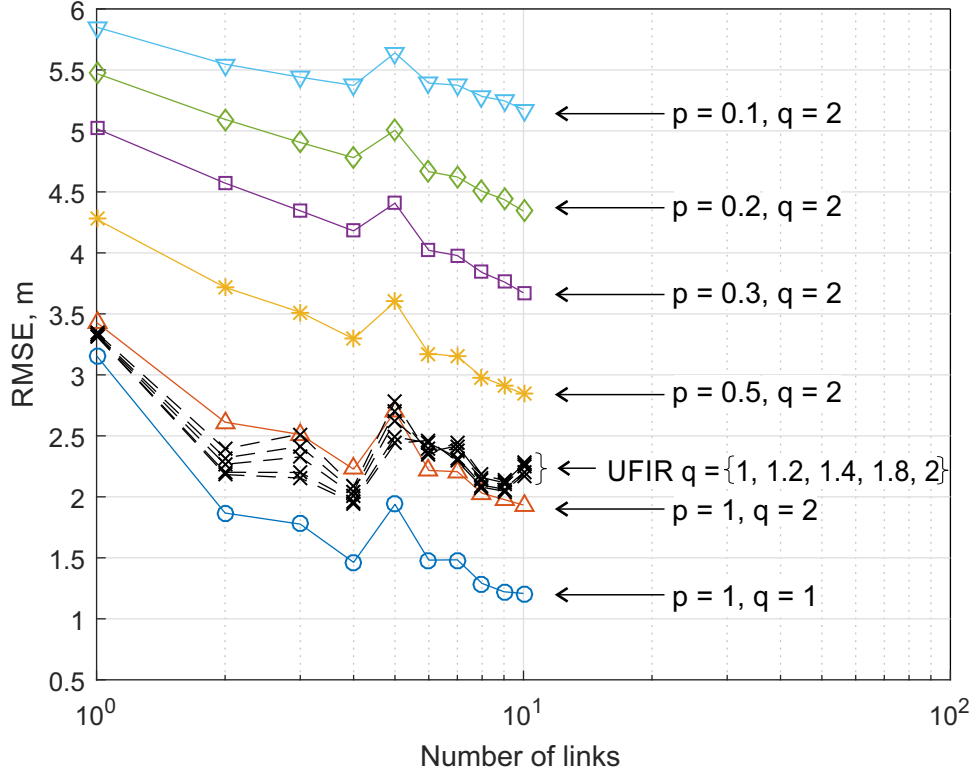


Figure 5.4: Root MSE (RMSEs) as functions of the number of links produced by the dUFIR filter and dKF under $p \neq 1$ and $q \neq 1$. Factor λ_k^{opt} of the dUFIR filter is affected by $q = \{1, 1.2, 1.4, 1.8, 2\}$.

$E\{\Delta y_k^{(i)2}\}$, ignore products of small and uncorrelated $\Delta \rho_k^{(i)}$ and $\Delta \phi_k^{(i)}$, provide

$$\begin{aligned} \sigma_{xk}^{(i)2} &= \sigma_{\rho k}^2 \cos^2 \bar{\phi}_k + \sigma_{\phi k}^2 \bar{\rho}_k^2 \sin^2 \bar{\phi}_k \\ &= \frac{(\Delta \rho_k^{(i)})^2}{9} \cos^2 \bar{\phi}_k + \frac{(\Delta \phi_k^{(i)})^2}{9} \bar{\rho}_k^2 \sin^2 \bar{\phi}_k \end{aligned} \quad (5.35)$$

$$\begin{aligned} \sigma_{yk}^{(i)2} &= \sigma_{\rho k}^2 \sin^2 \bar{\phi}_k + \sigma_{\phi k}^2 \bar{\rho}_k^2 \cos^2 \bar{\phi}_k \\ &= \frac{(\Delta \rho_k^{(i)})^2}{9} \sin^2 \bar{\phi}_k + \frac{(\Delta \phi_k^{(i)})^2}{9} \bar{\rho}_k^2 \cos^2 \bar{\phi}_k, \end{aligned} \quad (5.36)$$

and specify the measurement noise covariance with

$$R_k^{(i)} = \begin{bmatrix} \sigma_{xk}^{(i)2} & 0 \\ 0 & \sigma_{yk}^{(i)2} \end{bmatrix}.$$

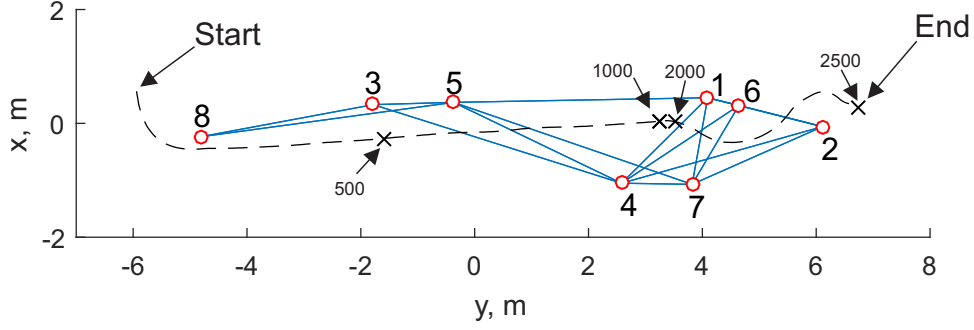


Figure 5.5: A WSN covering a part of the UGV ground truth trajectory (dashed).

Next, we represent the UVG dynamics in state space with

$$A = \begin{bmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H^{(i)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} \tau/2 & 0 \\ 1 & 0 \\ 0 & \tau/2 \\ 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} \sigma_{w2}^2 & 0 \\ 0 & \sigma_{w4}^2 \end{bmatrix},$$

where σ_{w2}^2 and σ_{w4}^2 are variances in the second and fourth states (velocities along x and y, respectively). For the average UVG velocity of 5.4 km/hour in the ground truth, we accept errors of 20 %, which gives $\sigma_{w2}^2 = \sigma_{w4}^2 = 0.09 \text{ m}^2/\text{s}^2$, and find $N_{\text{opt}} = 22$. Nodes available for the UVG at each time index k due to limited range are listed in Table 5.2.

Trajectory estimation via the WSN (Fig. 5.6) is obtained by combining estimates by nodes 8, 3, 5, 4, 6 and 2 (bolded in Table 5.2), which communicate with nearest neighbors (not bolded in Table 5.2). To test the algorithms, we consider several possible scenarios with a purpose of tuning filters as accurately as possible. To this end, we learn effects of deviations from the values $\Delta\rho_k = 4.8 \text{ cm}$ and $\Delta\phi_k = 0.47^\circ$ specified in the maximum sense on the filter performance via (5.35) and (5.36). In the first four scenarios, the dUFIR filter is tuned for $N_{\text{opt}} = 22$. In the remaining ones, N_{opt} is set individually to each filter. The scenarios are the following:

Table 5.2: Nodes Available in Different Time Intervals of Index k

Time index k	Node index
1-300	8
301-378	3,8
379-437	3
438-674	3,5
675-731	5
732-810	4,5
811-856	4
857-859	4,7
860-901	4,7
902-2154	1,4,6,7
2155-2179	1,2,4,6,7
2180-2254	1,2,6,7
2255-2319	1,2,6
2320-2387	2,6
2388-3000	2

1. *SC-1*: Set $\Delta\rho_k^{(i)} = 4.8$ cm and $\Delta\phi_k^{(i)} = 0.47^\circ$.
2. *SC-2*: Reduce $\Delta\rho_k^{(i)}$ and $\Delta\phi_k^{(i)}$ by the factor of 3 to be closer to actual sensor noise.
3. *SC-3*: Distribute sensor errors uniformly as $\Delta\rho_k^{(i)} \sim U(3.6, 4.8)$, in cm, and $\Delta\phi_k^{(i)} \sim U(0.1173^\circ, 0.47^\circ)$, but set $\Delta\rho_k^{(i)} = 4.8$ cm and $\Delta\phi_k^{(i)} = 0.47^\circ$.
4. *SC-4*: Reduce $\Delta\rho_k^{(i)} \sim U(3.6, 4.8)$, in cm, and $\Delta\phi_k^{(i)} \sim U(0.1173^\circ, 0.47^\circ)$ by the factor of 3 and set $\Delta\rho_k^{(i)} = 4.8$ cm and $\Delta\phi_k^{(i)} = 0.47^\circ$.
5. *SC-5*: Set $\Delta\rho_k^{(i)}$ randomly taken from $U(3.6, 4.8)$, in cm, and $\Delta\phi_k^{(i)}$ from $U(0.1173^\circ, 0.47^\circ)$. Set $N_{\text{opt}}^{(i)}$ individually to each sensor.
6. *SC-6*: Consider SC-5 for $Q \leftarrow p^2 Q_k$ with $p = 3$.

The absolute estimation errors are sketched in Fig. 5.6 along the coordinate x and one can easily trace the differences. It follows that the dUFIR filter generally outperforms the

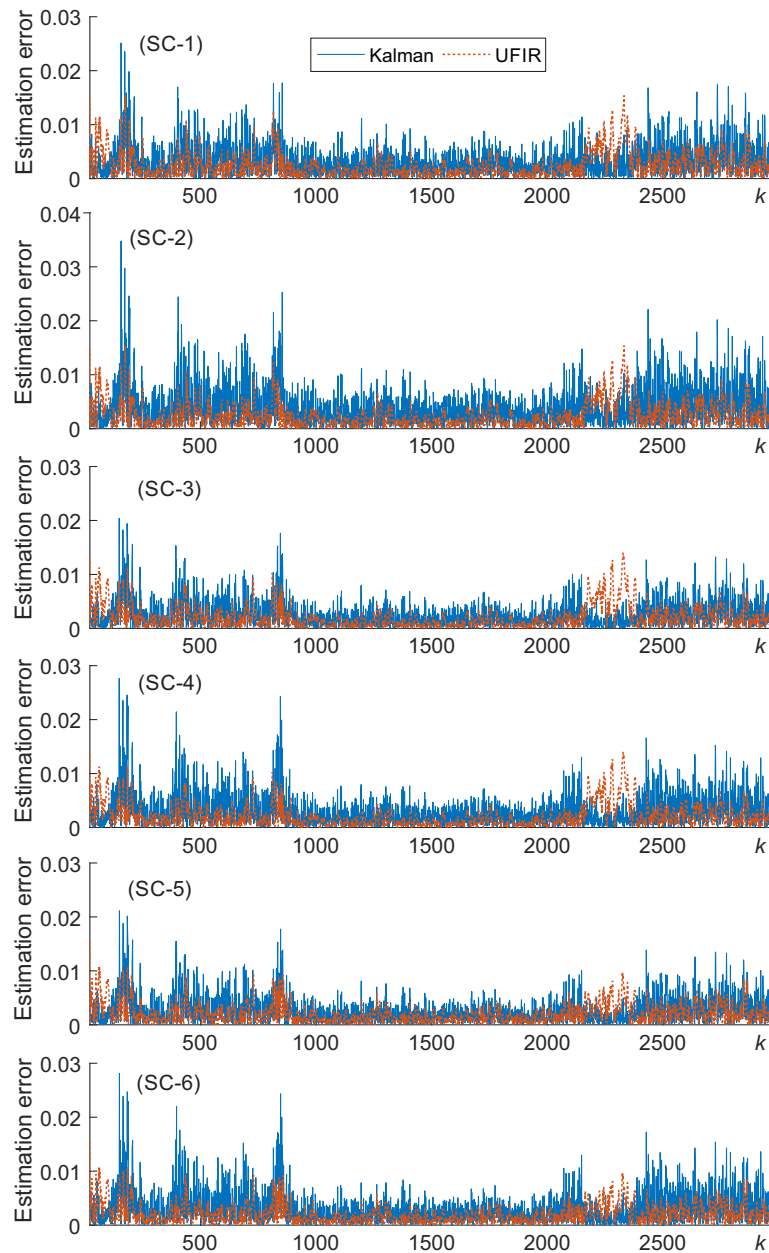


Figure 5.6: Absolute estimation errors along the coordinate x produced by the dKF and dUFIR filter for six scenarios, (SC-1)–(SC-6).

dKF in each of the above scenarios. To support this inference, the RMSEs computed by the root square of the sum of the MSEs along coordinates x and y are listed in Table 5.3, where the minimum values are bolded. It is also seen that errors in both filters become minimal

Table 5.3: RMSEs Produced by dKF and dUFIR Filter

Scenario	dKF	dUFIR
SC1	0.0077	0.0059
SC2	0.0092	0.0052
SC3	0.0063	0.0052
SC4	0.0076	0.0052
SC5	0.0065	0.0052
SC6	0.0065	0.0050

when the UGV moves very slowly ($1000 \leq k \leq 2000$) and grow otherwise. Furthermore, any change in the speed results in excursions, as at the beginning part of the trajectory, $0 \leq k \leq 900$.

A less favorable case for the dUFIR filter is when the UGV crosses straight lines connecting nodes 3 and 8 when $301 \leq k \leq 378$ and 1, 2, and 6 when $2255 \leq k \leq 2319$. Here, the dUFIR filter produces larger errors owing to the viewing angle φ of about π . For such an unilateral nodes deployments, the optimal φ is known to be $\pi/2$ [132] and errors grow essentially when $\varphi \rightarrow 0$ and $\varphi \rightarrow \pi$ [133]. Surprisingly, the dKF is less sensitive to φ in this area that needs further investigations. However, this case can easily be avoided in WSNs with bilateral nodes deployment.

We finally measure the computation time required by the batch and iterative algorithms as shown in Fig. 5.7 for a WSN organized with one dummy node and 1-9 neighbors. We provide it using Matlab R2016a operating on Intel Core i5 CPU(2.5 GHz) with 4.00 GB RAM under a macOS HighSierra ver. 10.13.6. As can be seen, Algorithm 5 operates faster than others and is almost invariant to the number of neighbors.

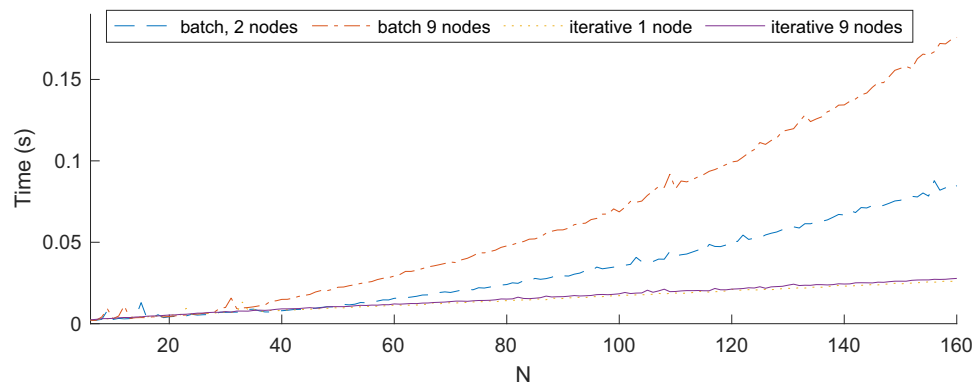


Figure 5.7: Measured computation time consumed by Algorithm 5 and batch filters (5.15) and (5.20) on different horizons N .

Chapter 6

Design of UFIR Filter With Consensus on Estimates for unstable WSNs

As demonstrated in the previous chapter, the consensus and optimization factor λ_k allows us to reduce the RMSE of the estimation while showing better robustness than KF against unexpected model errors, noise uncertainties and time-varying noise statistics. Despite the good results, the algorithm was tested assuming stable WSNs. In a real life scenario, WSNs are subject to many external factors that will cause missing data. In this chapter, we further our research by implementing a prediction feature on the dUFIR filter which allows to obtain robust estimates against missing measurements.

6.1 Introduction

Target tracking of moving objects is an application that benefits from unique advantages of wireless sensor networks (WSNs) [134, 135, 136] such as a massive nodes deployment, capacity of distributed processing, and ubiquitous integration with the environment. An example of indoor target tracking is shown in Fig. 6.1, where a WSN covers the moving object trajectory. A specific is that, due to the WSN restrictions, algorithms capable of estimating the position of a mobile object must comply with a sufficient accuracy and

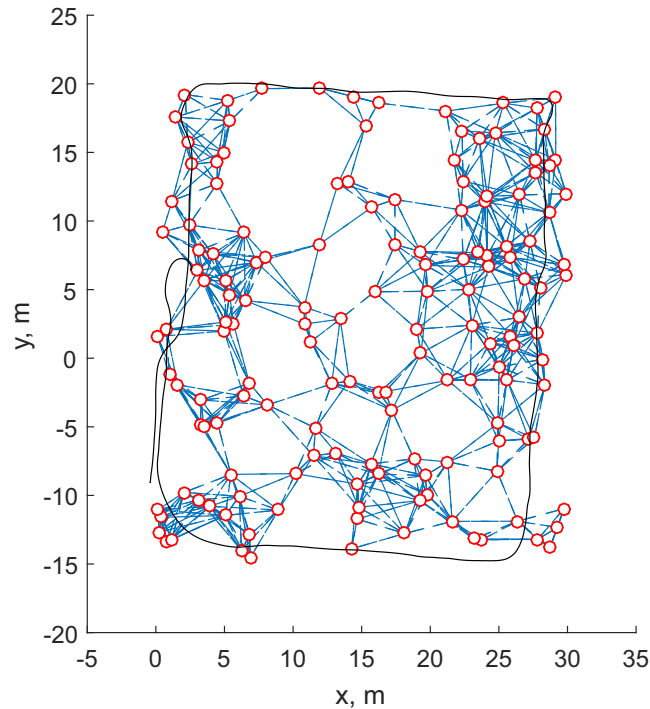


Figure 6.1: An example of a WSN composed of 150 nodes with 687 links (dotted). The WSN covers the moving object trajectory (solid) with the number of the links limited with the nodes ranges.

robustness required to provide tracking in the presence of model errors, missing data, and not completely known noise statistics. Therefore, optimal estimators and fusion techniques taking advantages of redundant and distributed measurements are often used to provide best noise reduction for WSN structures [15, 11, 16, 17, 18, 19, 20, 21].

For target tracking, the WSNs can be organized to have either a centralized, decentralized, or distributed structure [11]. The latter is known to be most powerful, flexible, and energy efficient [22, 23, 24]. Furthermore, it may provide even better estimates if to take advantage of different kinds of consensuses such as on measurements [26], estimates [1], information matrices [40], or other dynamic features [122, 123].

Among possible fusion techniques, the Kalman filter (KF)-based estimator remain most popular due to simplicity, optimality, and low computational burden [30, 31, 126, 137]. However, it is known that the optimality does not always go along with the robustness and fault tolerance required by the WSN operation conditions. The problem is that optimal estimators require all information about an object and its measurement, which is typically

unavailable in practice [42, 43].

Another issue is that measurements via WSNs are often accompanied with missing data due to external factors such as electromagnetic interference, unstable links, faulty behavior of the sensors, etc. [138]. Therefore, an algorithm must be capable of providing accurate estimation under temporary lost data as shown in many papers. For example, the state error covariance is bounded in [139] by introducing a critical value for the data arrival rate. In [30], the issue was solved by combining node estimates at the previous and current time points. The problem complicates by the fact that the KF estimate is affected by model errors and inappropriate noise behavior. Moreover, errors caused by missed data propagate along the entire estimation process.

As an alternative to the KF, there was developed a more robust approach employing properties of finite impulse response (FIR) filtering [43, 10]. Based upon this approach, the authors of [31] proposed a fusion technique using the optimal unbiased FIR (OUFIR) filter, which is more robust than the KF. In [21, 130], different types of consensus were taken into account using advantages of the unbiased FIR (UFIR) filter, which performs better than the KF under the real world operating conditions. Nevertheless, still no UFIR solution was addressed to designers of distributed WSNs with missing data that motivates our present work.

In this chapter, we design a distributed UFIR (dUFIR) filter for object tracking via WSNs with consensus on estimates under measurements with missing data. We show that the dUFIR filter outperforms the distributed KF (dKF) in terms of accuracy and robustness.

6.2 Moving Object Model in Distributed WSNs and Problem Formulation

In the state-space formulation, dynamics of a moving object can be described in discrete time index k and K -state space for a general scenario of distributed WSNs with missing

data using the following equations,

$$x_k = F_k x_{k-1} + B_k w_k, \quad (6.1)$$

$$\bar{y}_k^{(i)} = H_k^{(i)} F_k x_{k-1}, \quad (6.2)$$

$$y_k^{(i)} = \gamma_k (H_k^{(i)} x_k + v_k^{(i)}) + (1 - \gamma_k) \bar{y}_k^{(i)}, \quad (6.3)$$

$$y_k = H_k x_k + v_k, \quad (6.4)$$

where $x_k \in \mathbb{R}^K$, $F_k \in \mathbb{R}^{K \times K}$, and $B_k \in \mathbb{R}^{K \times L}$. Measurements of \mathcal{Q}_k are provided at each k with $J \triangleq J_k$ nodes. The i th, $i \in [1, J]$, node measures x_k by $y_k^{(i)} \in \mathbb{R}^p$, $p \leq K$, with $H_k^{(i)} \in \mathbb{R}^{p \times K}$ and each node has J inclusive neighbors. Local data $y_k^{(i)}$ are united in the observation vector $y_k = [y_k^{(1)T} \dots y_k^{(J)T}]^T \in \mathbb{R}^{Jp}$ with $H_k = [H_k^{(1)T} \dots H_k^{(J)T}]^T \in \mathbb{R}^{Jp \times K}$. Noise vectors $w_k \in \mathbb{R}^L$ and $v_k = [v_k^{(1)T} \dots v_k^{(n)T}]^T \in \mathbb{R}^{Jp}$ are zero mean, not obligatorily white Gaussian, uncorrelated, and with the covariances $Q_k = E\{w_k w_k^T\} \in \mathbb{R}^{L \times L}$, $R_k = \text{diag}[R_k^{(1)T} \dots R_k^{(n)T}]^T \in \mathbb{R}^{Jp \times Jp}$, and $R_k^{(i)} = E\{v_k^{(i)} v_k^{(i)T}\}$.

A binary variable γ_k serves as an indicator of whether data exist ($\gamma_k = 1$) or not ($\gamma_k = 0$). In the following section, we present the design of a batch dUFIR filter with optimal consensus on estimates that minimizes the mean square error (MSE) and show that it outperforms the dKF in terms of the localization robustness for measurements with missing data.

6.3 Tracking Filtering Algorithms with Consensus on Estimates

If to regard a WSN as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each vertex $v^{(i)} \in \mathcal{V}$ is a node and each link is an edge of set \mathcal{E} , for $i \in \mathcal{I} = \{1, \dots, n\}$ and $n = |\mathcal{V}|$. As stated in [25], nodes $v^{(i)}$ and $v^{(j)}$ reach an agreement if and only if states are related as $x^{(i)} = x^{(j)}$, $\{i, j\} \in \mathcal{I}$, $i \neq j$. Under such a condition, the network reaches a consensus with the common value of all nodes called the *group decision value*.

For the nodes to reach an agreement, a consensus protocol must minimize the total disagreement in the network by minimizing the Laplacian potential of the graph $\Psi = \frac{1}{2} x^T L x$, where L is the Laplacian matrix. A known linear distributed protocol for minimizing the

total disagreement is

$$u^{(i)} = \sum_j^J (x^{(j)} - x^{(i)}). \quad (6.5)$$

In what follows, we achieve the consensus of estimates by implementing (6.5) in two different algorithms: one based on the dKF and the other one on the dUFIR filter. The dKF requires that every first order neighbor shares the estimate of a local KF and that the i th node implements another KF with the consensus protocol (6.5) to reach the group decision value. In the dUFIR, the consensus on estimates is achieved using data only of the inclusive neighbors in (6.5). Details of the designed algorithms follow next.

6.3.1 Distributed KF Algorithm

The dKF with consensus on estimates was proposed in [1]. An idea behind this solution is to provide individual estimates in each node using the KF and then use another KF to fuse them. A pseudo code of the dKF algorithm augmented with a prediction option for temporary lost data is shown as Algorithm 7. Its specific is that, in order to reach a consensus on estimate when some data are temporary lost, an unavailable measurement at k is predicted (lines 3–5) via the available estimate $\bar{x}_k^{(j)}$ at $k - 1$ in each of the nodes. Because all data are needed from all of the neighbors, the dKF algorithm must ensure that the prediction is available from all of the neighbors.

6.3.2 Distributed UFIR Filter Algorithm

Unlike the dKF, which operates from one point to another using optimal recursions, the UFIR filter operates on finite horizons of N points and therefore exists in the convolution-based batch form and fast iterative form using recursions. Below, we show both these forms.

Algorithm 7: Iterative dKF Algorithm**Data:** $P_0^{(i)}, Q_k, R_k^{(j)}, \bar{x}_k^{(j)}, y_k^{(j)}, \bar{x}_0^{(i)} = x_0$ **Result:** $\hat{x}_k^{(i)}$ 1 **begin**2 **for** $k = 0 : \infty$ **do**3 **if** $\gamma_k = 0$ **then**4 $y_k^{(j)} = H_k^{(j)} F_k \hat{x}_{k-1}^{(j)}$;5 **end if**6 $z_k^{(j)} = H_k^{(j)T} R_k^{(j)-1} y_k^{(j)}, \quad \forall j \in J$;7 $s_k^{(i)} = \sum_{j \in J} z_k^{(j)}$;8 $Z_k^{(j)} = H_k^{(j)T} R_k^{(j)-1} H_k^{(j)}, \quad \forall j \in J$;9 $S_k^{(i)} = \sum_{j \in J} Z_k^{(j)}$;10 $M_k^{(i)} = (P_k^{(i)-1} + S_k^{(i)})^{-1}$;11 $\hat{x}_k^{(i)} = \bar{x}_k^{(i)} + M_k^{(i)} (s_k^{(i)} - S_k^{(i)} \bar{x}_k^{(i)}) + \epsilon M_k^{(i)} \sum_{j \in J} (\bar{x}_k^{(j)} - \bar{x}_k^{(i)})$;12 $P_k^{(i)} \leftarrow F_k M_k^{(i)} F_k^T + B_k Q_k B_k^T$;13 $\bar{x}_k^{(i)} \leftarrow F_k \hat{x}_k^{(i)}$;14 **end for**15 **end****Extended State-Space Model**

To apply FIR filtering, model (6.1)–(6.4) for $\gamma_k = 1$ must be extended on a horizon $[m, k]$ of N points, from $m = k - N + 1$ to k , as in the following [10],

$$X_{m,k} = A_{m,k} x_m + D_{m,k} W_{m,k}, \quad (6.6)$$

$$Y_{m,k} = C_{m,k} x_m + M_{m,k} W_{m,k} + V_{m,k}, \quad (6.7)$$

$$Y_{m,k}^{(i)} = C_{m,k}^{(i)} x_m + M_{m,k}^{(i)} W_{m,k} + V_{m,k}^{(i)}, \quad (6.8)$$

where $X_{m,k} = [x_m^T \ x_{m+1}^T \ \dots \ x_k^T]^T$, $Y_{m,k} = [y_m^T \ y_{m+1}^T \ \dots \ y_k^T]^T$, $W_{m,k} = [w_m^T \ w_{m+1}^T \ \dots \ w_k^T]^T$, $V_{m,k} = [v_m^T \ v_{m+1}^T \ \dots \ v_k^T]^T$, $Y_{m,k}^{(i)} = [y_m^{(i)T} \ y_{m+1}^{(i)T} \ \dots \ y_k^{(i)T}]^T$, $V_{m,k}^{(i)} = [v_m^{(i)T} \ v_{m+1}^{(i)T} \ \dots \ v_k^{(i)T}]^T$,

and the extended matrices are

$$A_{m,k} = [I \ F_{m+1}^T \ \dots \ (\mathcal{F}_{k-1}^{m+1})^T]^T, \quad (6.9)$$

$$D_{m,k} = \begin{bmatrix} B_m & 0 & \dots & 0 & 0 \\ F_{m+1}B_m & B_{m+1} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathcal{F}_{k-1}^{m+1}B_m & \mathcal{F}_{k-1}^{m+2}B_{m+1} & \dots & B_{k-1} & 0 \\ \mathcal{F}_k^{m+1}B_m & \mathcal{F}_k^{m+2}B_{m+1} & \dots & F_k B_{k-1} & B_k \end{bmatrix}, \quad (6.10)$$

$C_{m,k} = \bar{C}_{m,k}A_{m,k}$, $M_{m,k} = \bar{C}_{m,k}D_{m,k}$, $C_{m,k}^{(i)} = \bar{C}_{m,k}^{(i)}A_{m,k}$, $M_{m,k}^{(i)} = \bar{C}_{m,k}^{(i)}D_{m,k}$, where

$$\bar{C}_{m,k} = \text{diag}(H_m H_{m+1} \dots H_k), \quad (6.11)$$

$$\bar{C}_{m,k}^{(i)} = \text{diag}(H_m^{(i)} H_{m+1}^{(i)} \dots H_k^{(i)}), \quad (6.12)$$

$$\mathcal{F}_r^g = \begin{cases} F_r F_{r-1} \dots F_g, & g < r+1 \\ I, & g = r+1 \\ 0, & g > r+1 \end{cases}. \quad (6.13)$$

Based on model (6.6)–(6.8), the batch dUFIR filter can be designed as shown below.

Batch dUFIR Filter

The FIR estimate for model (6.6)–(6.8) can be obtained as

$$\hat{x}_k = \Theta_{m,k} Y_{m,k}, \quad (6.14)$$

where $\Theta_{m,k}$ is the FIR filter gain (impulse response) obeying some cost function [10]. To obtain the dUFIR filter, let us suppose that the i th node provides a local estimate over data (6.8) as $\hat{x}_k^{(i)}$. Then, referring to [1], the consensus between the local estimates can be found if to introduce a vector $\Sigma_k = \sum_j^J [\hat{x}_k^{(j)} - \hat{x}_k^{(i)}]$, combine it with (6.14), and write the estimate as

$$\hat{x}_k^c = \Theta_{m,k} Y_{m,k} + \lambda_k \Sigma_k, \quad (6.15)$$

where λ_k is a scaling factor to be optimized in the MSE sense.

For the dUFIR filter, gain $\Theta_{m,k}$ must be found to obey the unbiased condition

$$E\{\hat{x}_k^c\} = E\{\hat{x}_k^{(i)}\} = E\{x_k\}$$

and the dUFIR estimate will thus be robust against errors in the noise statistics and initial values [10].

Referring to (6.6)–(6.8), estimate (6.15) can be rewritten as

$$\hat{x}_k^c = \Theta_{m,k} Y_{m,k} + J\lambda_k \Theta_{m,k} Y_{m,k} - J\lambda_k \Theta_{m,k}^{(i)} Y_{m,k}^{(i)}, \quad (6.16)$$

where gains $\Theta_{m,k}$ and $\Theta_{m,k}^{(i)}$, which obey the unbiasedness condition [10], are represented with

$$\Theta_{m,k} = (I + J\lambda_k)(\mathcal{H}_{m,k}^T \mathcal{H}_{m,k})^{-1} \mathcal{H}_{m,k}^T, \quad (6.17a)$$

$$= (I + J\lambda_k) G_k \mathcal{H}_{m,k}^T, \quad (6.17b)$$

$$\Theta_{m,k}^{(i)} = (\mathcal{H}_{m,k}^{(i)T} \mathcal{H}_{m,k}^{(i)})^{-1} \mathcal{H}_{m,k}^{(i)T}, \quad (6.18a)$$

$$= G_k^{(i)} \mathcal{H}_{m,k}^{(i)T}, \quad (6.18b)$$

where $G_k^{(i)} = (\mathcal{H}_{m,k}^{(i)T} \mathcal{H}_{m,k}^{(i)})^{-1}$ is the generalized noise power gain (GNPG) [52], and

$$\mathcal{H}_{m,k} = \begin{bmatrix} H_m (\mathcal{F}_k^{m+1})^{-1} \\ H_{m+1} (\mathcal{F}_k^{m+2})^{-1} \\ \vdots \\ H_{k-1} F_k^{-1} \\ H_k \end{bmatrix}, \quad (6.19)$$

$$\mathcal{H}_{m,k}^{(i)} = \begin{bmatrix} H_m^{(i)} (\mathcal{F}_k^{m+1})^{-1} \\ H_{m+1}^{(i)} (\mathcal{F}_k^{m+2})^{-1} \\ \vdots \\ H_{k-1}^{(i)} F_k^{-1} \\ H_k^{(i)} \end{bmatrix}. \quad (6.20)$$

As can be seen, information required to compute $\Theta_{m,k}$ and $\Theta_{m,k}^{(i)}$ is entirely provided by the K -state space model, which can be preloaded on the nodes. Thus, only measurement data will be sent by the node, unlike the dKF case implying that each node must wait for the individual estimate of its neighbors. This reduces the number of exchange messages and improves battery life.

The optimal scaling factor λ_k^{opt} can be obtained by solving the optimization problem

$$\lambda_k^{\text{opt}} = \arg \min_{\lambda_k} \{ \text{tr } P_k(\lambda_k) \},$$

where $P_k = E\{\varepsilon_k \varepsilon_k^T\}$ is the error covariance matrix and $\varepsilon_k = x_k - \hat{x}_k^c$ is the estimation error. By solving the optimization problem, λ_k^{opt} can be shown to be (see Appendix A)

$$\begin{aligned} \lambda_k^{\text{opt}} &= -\frac{1}{J} (\tilde{\Theta}_{m,k} \bar{R}_{m,k} \tilde{\Theta}_{m,k}^T - G_k G_k^{(i)-1} \Theta_{m,k}^{(i)} \\ &\quad \times \bar{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T}) (\tilde{\Theta}_{m,k} \bar{R}_{m,k} \tilde{\Theta}_{m,k}^T - 2G_k G_k^{(i)-1} \\ &\quad \times \Theta_{m,k}^{(i)} \bar{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T} + \Theta_{m,k}^{(i)} \bar{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T})^{-1}. \end{aligned} \quad (6.21)$$

where $\tilde{\Theta}_{m,k} = G_k \mathcal{H}_{m,k}^T$ and

$$\bar{R}_{m,k} = E\{v_{m,k} v_{m,k}^T\} = \text{diag}(R_m \dots R_k), \quad (6.22)$$

$$\bar{R}_{m,k}^{(i)} = E\{v_{m,k}^{(i)} v_{m,k}^{(i)T}\} = \text{diag}(R_m^{(i)} \dots R_k^{(i)}). \quad (6.23)$$

Although (6.21) is affected by measurement noise, it remains invariant to system noise, that definitely results in higher robustness of the dUFIR filter.

A flaw of the batch dUFIR filter is that the implementation of (6.15) with (6.21) on a high-density WSN and large horizons N require a large-dimension matrix operation, which is not suitable for smart sensors. A fast computation can be provided using an iterative algorithm, which we will consider next.

Iterative dUFIR Filtering Algorithm

An iterative form of the estimate \hat{x}_k^c can be obtained if to represent \hat{x}_k^c with a sum of a centralized estimate \hat{x}_k defined by (6.14) and a local estimate $\hat{x}_k^{(i)} = \Theta_{m,k}^{(i)} Y_{m,k}^{(i)}$. That allows writing (6.16) as

$$\hat{x}_k^c = (I + J\lambda_k) \hat{x}_k - J\lambda_k \hat{x}_k^{(i)}, \quad (6.24)$$

and, following [10, 21], find recursions. Namely, for \hat{x}_k , one can employ from [10]

$$G_l = [H_l^T H_l + (F_l G_{l-1} F_l^T)^{-1}]^{-1}, \quad (6.25)$$

$$\hat{x}_l^- = F_l \hat{x}_{l-1}, \quad (6.26)$$

$$\hat{x}_l = \hat{x}_l^- + G_l H_l^T (y_l - H_l \hat{x}_l^-), \quad (6.27)$$

and

$$G_l^{(i)} = [H_l^{(i)T} H_l^{(i)} + (F_l G_{l-1}^{(i)} F_l^T)^{-1}]^{-1}, \quad (6.28)$$

$$\hat{x}_l^{(i)-} = F_l \hat{x}_{l-1}^{(i)}, \quad (6.29)$$

$$\hat{x}_l^{(i)} = \hat{x}_l^{(i)-} + G_l^{(i)} H_l^{(i)T} (y_l^{(i)} - H_l^{(i)} \hat{x}_l^{(i)-}), \quad (6.30)$$

where l is an iterative variable starting at $s = k - N + K$, where K is the number of the states, and ending when $l = k$.

Iterations using (6.25)–(6.27) can be initialized with $G_{l-1} = G_s$ and $\hat{x}_{l-1} = \hat{x}_s$ in short batch forms of

$$G_s = (\mathcal{H}_{m,s}^T \mathcal{H}_{m,s})^{-1}, \quad (6.31)$$

$$\hat{x}_s = G_s \mathcal{H}_{m,s}^T Y_{m,s}. \quad (6.32)$$

Following the same strategy, iterations (6.28)–(6.30) for $\hat{x}_k^{(i)}$ can be initialized with

$$G_s^{(i)} = (\mathcal{H}_{m,s}^{(i)T} \mathcal{H}_{m,s}^{(i)})^{-1}, \quad (6.33)$$

$$\hat{x}_s^{(i)} = G_s^{(i)} \mathcal{H}_{m,s}^{(i)T} Y_{m,s}^{(i)}. \quad (6.34)$$

Finally, fast computation of factor λ_k^{opt} can be provided if to represent (6.21) as

$$\lambda_k^{\text{opt}} = -\frac{1}{J} (\alpha_k - G_k G_k^{(i)-1} \beta_k) (\alpha_k - 2G_k G_k^{(i)-1} \beta_k + \beta_k)^{-1}, \quad (6.35)$$

where $\alpha_k = \tilde{\Theta}_{m,k} \bar{R}_{m,k} \tilde{\Theta}_{m,k}^T$ and $\beta_k = \Theta_{m,k}^{(i)} \bar{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T}$, and use the recursions (see Appendix A and Appendix A)

$$\alpha_k = G_k (H_k^T R_k H_k + F_k^{-T} G_{k-1}^{-1} \alpha_{k-1} G_{k-1}^{-1} F_k^{-1}) G_k \quad (6.36)$$

$$\beta_k = G_k^{(i)} (H_k^{(i)T} R_k^{(i)} H_k^{(i)} + F_k^{-T} G_{k-1}^{(i)-1} \beta_{k-1} G_{k-1}^{(i)-1} F_k^{-1}) G_k^{(i)}, \quad (6.37)$$

which initial values α_{k-1} and β_{k-1} can be computed in short batch forms as

$$\alpha_s = G_s \mathcal{H}_{m,s}^T \bar{R}_{m,s} \mathcal{H}_{m,s} G_s^T, \quad (6.38)$$

$$\beta_s = G_s^{(i)} \mathcal{H}_{m,s}^{(i)T} \bar{R}_{m,s}^{(i)} \mathcal{H}_{m,s}^{(i)} G_s^{(i)T}. \quad (6.39)$$

A pseudo code of the predictive iterative dUFIR algorithm with consensus on estimates designed for measurements with temporary missing data is listed as Algorithm 8. Given

Algorithm 8: Iterative dUFIR Filtering Algorithm**Data:** $y_k, R_k^{(i)}, R_k, N$ **Result:** \hat{x}_k 1 **begin**2 **for** $k = N - 1 : \infty$ **do**3 $m = k - N + 1, \quad s = m + K - 1;$ 4 $G_s = (\mathcal{H}_{m,s}^T \mathcal{H}_{m,s})^{-1};$ 5 $G_s^{(i)} = (\mathcal{H}_{m,s}^{(i)T} \mathcal{H}_{m,s}^{(i)})^{-1};$ 6 **if** $\gamma_k = 0$ **then**7 $y_k^{(j)} = H_k^{(j)} F_k \hat{x}_{k-1}^{(j)};$ 8 **end if**9 $\tilde{x}_s = G_s \mathcal{H}_{m,s}^T Y_{m,s};$ 10 $\tilde{x}_s^{(i)} = G_s^{(i)} \mathcal{H}_{m,s}^{(i)T} Y_{m,s}^{(i)};$ 11 $\alpha_s = G_s \mathcal{H}_{m,s}^T \bar{R}_{m,s} \mathcal{H}_{m,s} G_s^T;$ 12 $\beta_s = G_s^{(i)} \mathcal{H}_{m,s}^{(i)T} \bar{R}_{m,s}^{(i)} \mathcal{H}_{m,s}^{(i)} G_s^{(i)T};$ 13 **for** $l = s + 1 : k$ **do**14 $\hat{x}_l^- = F_l \hat{x}_{l-1};$ 15 $\hat{x}_l^{(i)-} = F_l \hat{x}_{l-1}^{(i)};$ 16 $G_l = [H_l^T H_l + (F_l G_{l-1} F_l^T)^{-1}]^{-1};$ 17 $G_l^{(i)} = [H_l^{(i)T} H_l^{(i)} + (F_l G_{l-1}^{(i)} F_l^T)^{-1}]^{-1};$ 18 $\hat{x}_l = \hat{x}_l^- + G_l H_l^T (y_l - H_l \hat{x}_l^-);$ 19 $\hat{x}_l^{(i)} = \hat{x}_l^{(i)-} + G_l^{(i)} H_l^{(i)T} (y_l^{(i)} - H_l^{(i)} \hat{x}_l^{(i)-});$ 20 $\alpha_l = G_l (H_l^T R_l H_l + F_l^{-T} G_{l-1}^{-1} \alpha_{l-1} G_{l-1}^{-1} F_l^{-1}) G_l;$ 21 $\beta_l = G_l^{(i)} (H_l^{(i)T} R_l^{(i)} H_l^{(i)} + F_l^{-T} G_{l-1}^{(i)-1} \beta_{l-1} G_{l-1}^{(i)-1} F_l^{-1}) G_l^{(i)};$ 22 **end for**23 $\lambda_k = -\frac{1}{j} (\alpha_k - G_k G_k^{(i)-1} \beta_k) (\alpha_k - 2G_k G_k^{(i)-1} \beta_k + \beta_k)$ 24 $\hat{x}_k^c = (I + J \lambda_k) \tilde{x}_k - J \lambda_k \tilde{x}_k^{(i)};$ 24 **end for**25 **end**26 † First data y_0, y_1, \dots, y_{N-1} must be available.

Table 6.1: Nodes Sorted by the Number of Available Links with Neighbors

	Number of links						
	2	3	4	5	6	7	9
Node	8,12,16	1	4,5,6,13,14,15	18	2,3,17	7,11	9,10

a horizon of N points, Algorithm 8 starts computing the initial values at $s = m + K - 1$ and then updates the results beginning at $s + 1$ until the iterative variable l reaches k . It then computes the optimal consensus factor λ_k and finishes with the output estimate \hat{x}_k^c .

In what follows, we will test Algorithm 8 along with the dKF Algorithm 7 originally proposed in [1]. A numerical example will be given for tracking of a circularly traveling and rapidly maneuvering object. Experimental verification will be provided for robot localization with measured ground truth.

6.4 Maneuvering Object Tracking with Missing Data

To conduct this experiment, we employ the ground truth trajectory available for free from the MagPIE project dataset [6]. We consider a random WSN composed of 18 nodes whose connections are sketched in Fig. 6.2 and listed in Table 6.1. Every node is capable of measuring the object Cartesian coordinates x and y of the location of the mobile robot. Measurements were simulated by adding white Gaussian noise to the ground truth data in each of the sensors. Noise was generated to have the variance $\sigma_v^{(i)} = 0.25 + \phi$, where ϕ is uniformly distributed as $\phi = U(0.5, -0.5)$.

Supposing that some data can be lost in communication channels, we remove some data obeying the binomial distribution with the probability of $P = 0.9$ as shown in Fig 6.3. Note that each node has different sets of lost data.

The moving object dynamics are given in state space by

$$A = \begin{bmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H^{(i)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

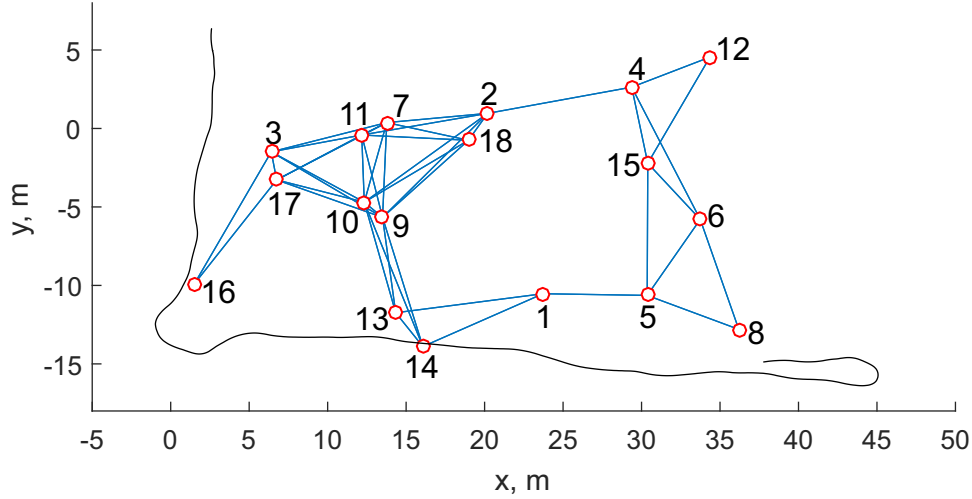


Figure 6.2: WSN over a ground truth trajectory available from the MagPIE project dataset [6].

$$B = \begin{bmatrix} \tau/2 & 0 \\ 1 & 0 \\ 0 & \tau/2 \\ 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_w^2 \end{bmatrix},$$

where $\sigma_w = 0.76$ m/s. For dUFIR, the optimal horizon N_{opt} was found at a test stage to be 53 in average.

As stated by (6.21), the optimal factor λ_k^{opt} depends on the appropriate knowledge of the noise statistics. To analyze the robustness of the algorithms, we let $Q_k \leftarrow (0.1)^2 Q_k$ and $R_k^{(i)} \leftarrow q^{(i)^2} R_k^{(i)}$ where $q^{(i)} = U(1, 2)$, meaning that each sensor has different errors in the noise statistics.

As has been shown in [21], the estimation error decreases by an increase in the number of the links. As follows from Fig. 6.4 sketching the RMSE produced by each node, this also holds true for the consensus on estimates. In fact, despite the effect of noise uncertainties in (6.20), the dUFIR filter errors range in Fig. 6.4 much lower than by the dKF. Also, the dUFIR filter demonstrates lesser variations in the individual RMSEs. The latter means that the dUFIR filter provided a better consensus than the dKF. Effect of errors in the noise covariance on the dKF estimate is easily seen in Fig. 6.5. Due to noise uncertainties, dKF fails to maintain low estimation errors, which is more evident as the number of neighbors

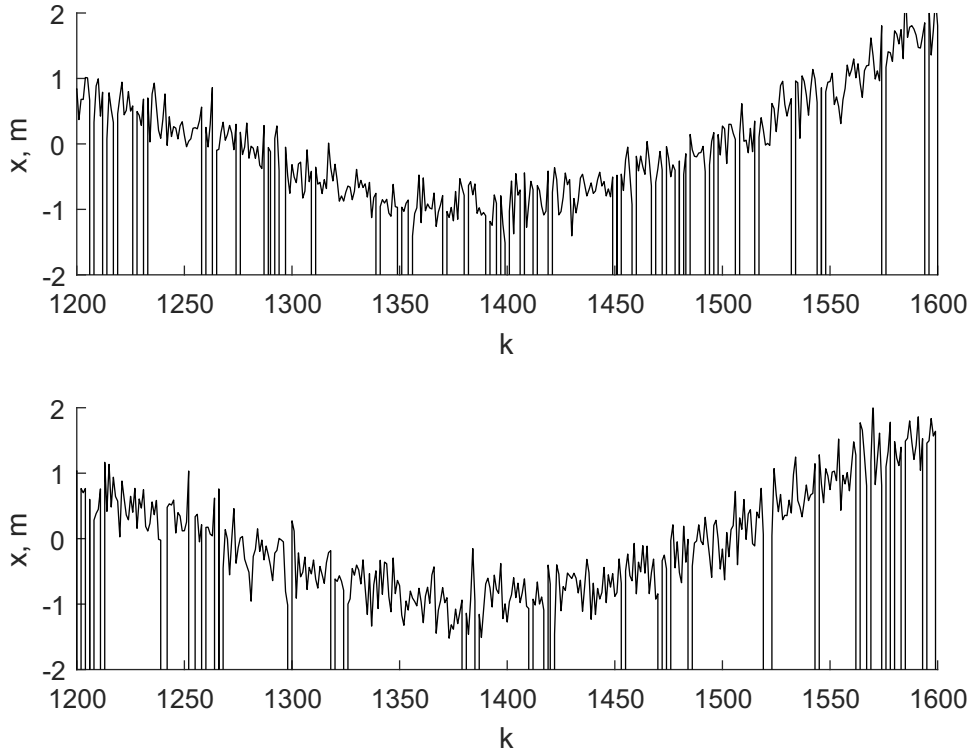


Figure 6.3: Measurements along coordinate x with missing data for: a) node 10 and b) node 12.

is small (Fig. 6.5 a). However, even when the number of neighbors increases, dUFIR still outperforms dKF (Fig. 6.5 b).

6.5 Vehicle Localization over WSN with Missing Data and Time-Varying Model Noise

In this section we consider a WSN with 30 nodes, which covers a trajectory of an unmanned ground vehicle (UGV) (robot). The trajectory shown in Fig. 6.6 is available for free use from the MagPIE dataset [6]. Each node is equipped with a time-of-flight (ToF) ranging sensor VL53L0X and a MEMS gyro ADXRS649. The measuring distance $\rho_k^{(i)}$ of the i th ToF sensor is limited with 2 m and an accuracy of $\Delta\rho_k^{(i)} = 4.8$ cm. The MEMS gyro has an angular resolution of $\Delta\phi_k^{(i)} = 0.47^\circ$ for a measured angle $\phi_k^{(i)}$. The communication range of each node is limited with 5 m.

When an UGV enters in the node range, a distance and an angle are measured as

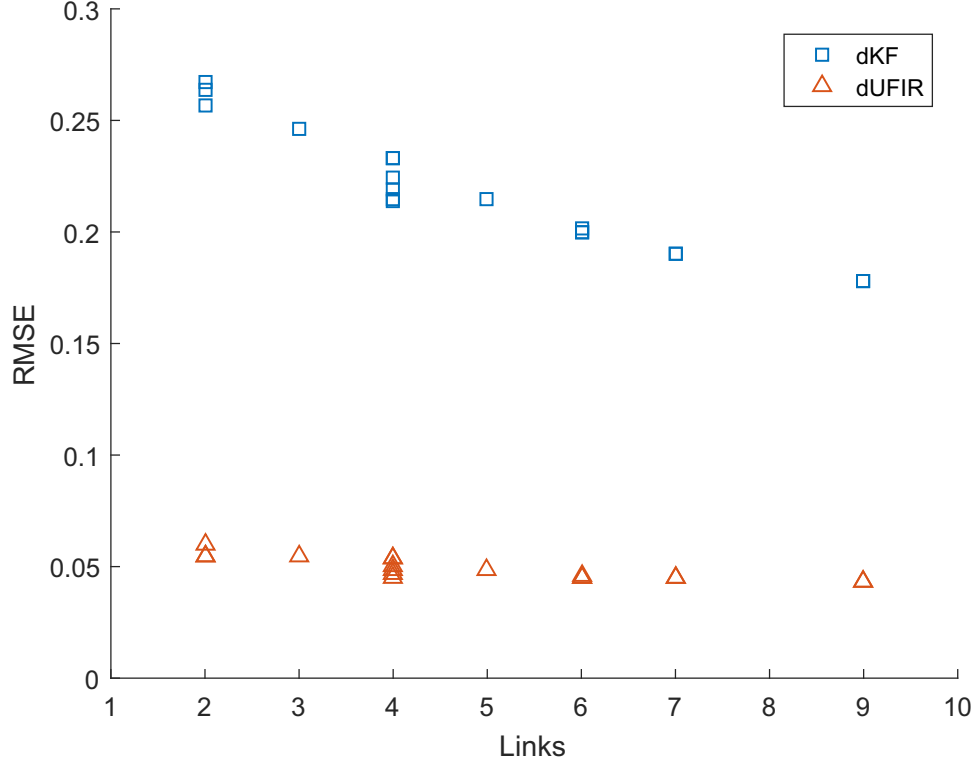


Figure 6.4: RMSE produced by each node of the WSN.

$\rho_k^{(i)} = \bar{\rho}_k^{(i)} + \Delta\rho_k^{(i)}$ and $\phi_k^{(i)} = \bar{\phi}_k^{(i)} + \Delta\phi_k^{(i)}$, respectively, with a sampling time of $T = 0.01s$, where $\bar{\rho}_k^{(i)}$ and $\bar{\phi}_k^{(i)}$ are average values and $\Delta\rho_k^{(i)}$ and $\Delta\phi_k^{(i)}$ are white Gaussian and uncorrelated with the standard deviations of $\sigma_{\rho k} = |\Delta\rho_k^{(i)}|/3$ and $\sigma_{\phi k} = |\Delta\phi_k^{(i)}|/3$. The UGV altitude is ignored in our experiment. To avoid nonlinearities inherent to polar coordinates, we represent the UGV Cartesian coordinates as $x_k^{(i)} = \rho_k^{(i)} \cos \phi_k^{(i)} = \bar{x}_k^{(i)} + \Delta x_k^{(i)}$ and $y_k^{(i)} = \rho_k^{(i)} \sin \phi_k^{(i)} = \bar{y}_k^{(i)} + \Delta y_k^{(i)}$ and approximate with

$$\begin{aligned} x_k^{(i)} &= (\bar{\rho}_k^{(i)} + \Delta\rho_k^{(i)}) \cos(\bar{\phi}_k^{(i)} + \Delta\phi_k^{(i)}), \\ &\cong \bar{\rho}_k^{(i)} \cos \bar{\phi}_k^{(i)} + \Delta\rho_k^{(i)} \cos \bar{\phi}_k^{(i)} - \Delta\phi_k^{(i)} \bar{\rho}_k^{(i)} \sin \bar{\phi}_k^{(i)}, \end{aligned} \quad (6.40)$$

$$\begin{aligned} y_k^{(i)} &= (\bar{\rho}_k^{(i)} + \Delta\rho_k^{(i)}) \sin(\bar{\phi}_k^{(i)} + \Delta\phi_k^{(i)}), \\ &\cong \bar{\rho}_k^{(i)} \sin \bar{\phi}_k^{(i)} + \Delta\rho_k^{(i)} \sin \bar{\phi}_k^{(i)} + \Delta\phi_k^{(i)} \bar{\rho}_k^{(i)} \cos \bar{\phi}_k^{(i)}, \end{aligned} \quad (6.41)$$

where $\bar{x}_k^{(i)} = \bar{\rho}_k^{(i)} \cos \bar{\phi}_k^{(i)}$, $\bar{y}_k^{(i)} = \bar{\rho}_k^{(i)} \sin \bar{\phi}_k^{(i)}$, $\Delta x_k^{(i)} = \Delta\rho_k^{(i)} \cos \bar{\phi}_k^{(i)} - \Delta\phi_k^{(i)} \bar{\rho}_k^{(i)} \sin \bar{\phi}_k^{(i)}$, and

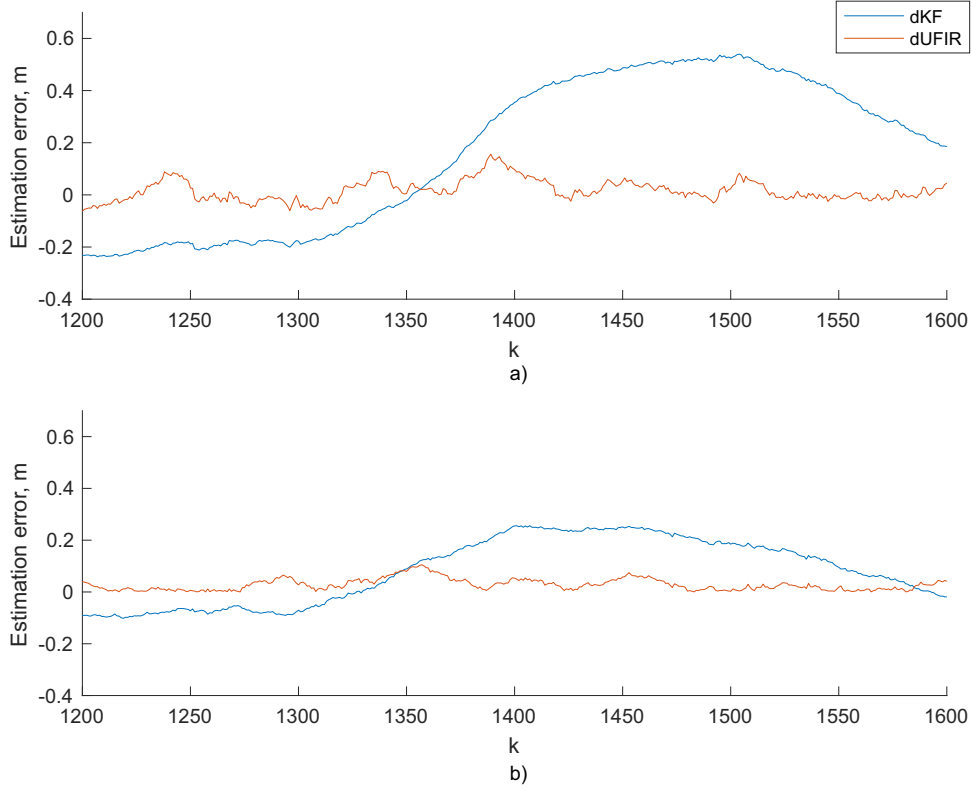


Figure 6.5: Estimation error produced by the dUFIR filter and dKF under error in the noise covariances: a) node 12 with 2 links and b) node 10 with 9 links.

$$\Delta y_k^{(i)} = \Delta \rho_k^{(i)} \sin \bar{\phi}_k^{(i)} + \Delta \phi_k^{(i)} \bar{\rho}_k^{(i)} \cos \bar{\phi}_k^{(i)}.$$

For this model, we define the measurement noise variances as $\sigma_{xk}^{(i)2} = E\{\Delta x_k^{(i)2}\}$ and $\sigma_{yk}^{(i)2} = E\{\Delta y_k^{(i)2}\}$, ignore products of small and uncorrelated values $\Delta \rho_k^{(i)}$ and $\Delta \phi_k^{(i)}$, provide

$$\sigma_{xk}^{(i)} = \sigma_{\rho k}^2 \cos^2 \bar{\phi}_k + \sigma_{\phi k}^2 \bar{\rho}_k^2 \sin^2 \bar{\phi}_k \quad (6.42)$$

$$\begin{aligned} &= \frac{(\Delta \rho_k^{(i)})^2}{9} \cos^2 \bar{\phi}_k + \frac{(\Delta \phi_k^{(i)})^2}{9} \bar{\rho}_k^2 \sin^2 \bar{\phi}_k \\ \sigma_{yk}^{(i)} &= \sigma_{\rho k}^2 \sin^2 \bar{\phi}_k + \sigma_{\phi k}^2 \bar{\rho}_k^2 \cos^2 \bar{\phi}_k \\ &= \frac{(\Delta \rho_k^{(i)})^2}{9} \sin^2 \bar{\phi}_k + \frac{(\Delta \phi_k^{(i)})^2}{9} \bar{\rho}_k^2 \cos^2 \bar{\phi}_k, \end{aligned} \quad (6.43)$$

and describe the time-varying measurement noise covariance matrix as

$$R_k^{(i)} = \begin{bmatrix} \sigma_{xk}^{(i)} & 0 \\ 0 & \sigma_{yk}^{(i)} \end{bmatrix}.$$

The UGV dynamics and the covariance Q are exactly the same as in the previous section.

The nodes available for the UGV at each k due to limited range are listed in Table 6.2.

Table 6.2: Nodes Available in Different Time Intervals of Index k

Time index k	Node index	Time index k	Node index
1 - 89	17,20,21	1208 -1287	22,26,27
90 - 90	20,21	1287 -1290	22,26
91 - 123	2,20,21	1290 -1324	5,22,26
124 - 165	2,21	1324 -1343	5,22
166 - 277	2	1343 -1404	5,22,30
278 - 300	2,11	1404 -1417	5,30
301 - 324	2,4,11	1417 -1425	5,25,30
325 - 397	4,11	1425 -1467	5,23,25,30
398 - 464	4,10,11	1467 -1476	23,25,30
465 - 481	10,11	1476 -1514	19,23,25,30
482 - 486	7,10,11	1514 -1534	19,23,30
487 - 502	7,10	1534 -1559	19,23
503 - 586	7,10,15	1559 -1602	12,19,23
587 - 624	7,15	1602 -1619	12,19,23,28
625 - 652	7,15,24	1619 -1681	12,19,28
653 - 668	7,24	1681 -1712	12,28
669 - 690	24	1712 -1714	12,14,28
691 - 739	18,24	1714 -1792	14,28
740 - 772	18,24,29	1792 -1817	14
773 - 797	1,18,24,29	1817 -1892	6,14
798 - 836	1,18,29	1892 -1946	6
837 - 898	1,29	1946 -1980	6,13
899 - 931	1,9,29	1980 -2037	13
932 - 962	1,9	2037 -2117	3,13
963 - 1066	9	2117 -2135	3
1067 - 1087	9,27	2135 -2200	3,16
1088 - 1105	27	2200 -2228	16
1106 - 1207	26,27	2228 -2300	8,16

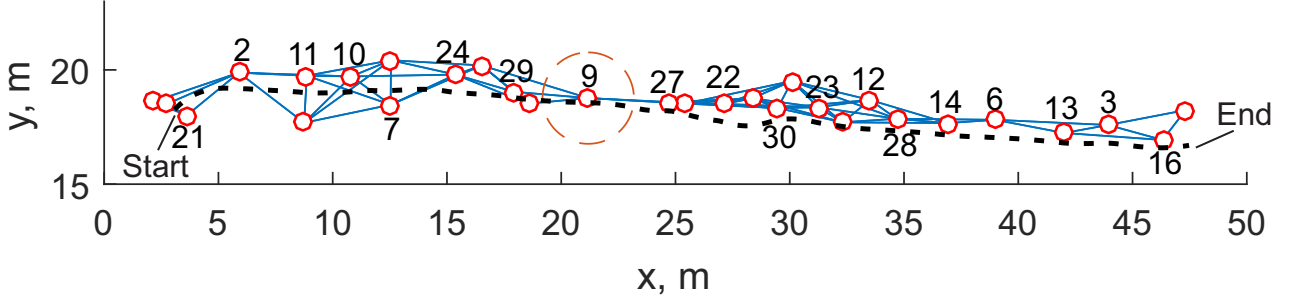


Figure 6.6: A WSN covering the UGV trajectory available from the MagPIE dataset [6]. The labeled nodes are used in the reconstruction of the trajectory. The dashed circle exhibits the 2 m range of a ToF sensor.

Estimation of the UGV trajectory via the WSN has been obtained by combining estimates by the nodes labeled in Fig. 6.6 and bolded in Table 6.2, which communicate with nearest neighbors (not bolded in Table 6.2). To test the algorithms for different available information about noise, we consider several possible scenarios of filter tuning. In each of the cases, we evaluate effects of deviations from $\Delta\rho_k = 4.8$ cm and $\Delta\phi_k = 0.47^\circ$ specified in the maximum sense on the filter performance via (6.42) and (6.43).

In the first and second scenarios, measurement data are simulated assuming that the normally distributed zero mean noise has the same variances of $(\Delta\rho_k^{(i)})^2 = 4.8^2$ and $(\Delta\phi_k^{(i)})^2 = 0.47^2$ for $i = \{1, \dots, 30\}$ in all sensors. In the remaining four scenarios, we generate different measurement data supposing that the normally distributed zero noise has different variances in each sensor. In this case, the variances are uniformly distributed with $(\Delta\rho_k^{(i)})^2 \sim U(3.6^2, 4.8^2)$ and $(\Delta\phi_k^{(i)})^2 \sim U(0.1173^2, 0.47^2)$ for $i = \{1, \dots, 30\}$. In the first four scenarios, the dUFIR filter is tuned to $N_{\text{opt}} = 13$. In the first five scenarios, the dKF undergoes the effect of errors in the noise statistics caused by $Q \leftarrow p^2 Q_k$ with $p = 4$. The scenarios are the following:

1. *SC-1*: Set $\Delta\rho_k^{(i)} = 4.8$ cm and $\Delta\phi_k^{(i)} = 0.47^\circ$.
2. *SC-2*: Reduce $\Delta\rho_k^{(i)}$ and $\Delta\phi_k^{(i)}$ by the factor of 3 as an error of the known sensor noise.
3. *SC-3*: Distribute sensor errors uniformly as $\Delta\rho_k^{(i)} \sim U(3.6, 4.8)$, in cm, and $\Delta\phi_k^{(i)} \sim U(0.1173^\circ, 0.47^\circ)$, but set $\Delta\rho_k^{(i)} = 4.8$ cm and $\Delta\phi_k^{(i)} = 0.47^\circ$.

Table 6.3: RMSEs Produced by dKF and dUFIR Filter

Scenario	dKF	dUFIR
SC1	0.0081	0.0077
SC2	0.0097	0.0077
SC3	0.0071	0.0070
SC4	0.0084	0.0070
SC5	0.0073	0.0069
SC6	0.0091	0.0069

4. *SC-4*: as in SC-3, distribute errors uniformly as $\Delta\rho_k^{(i)} \sim U(3.6, 4.8)$, in cm, and $\Delta\phi_k^{(i)} \sim U(0.1173^\circ, 0.47^\circ)$ and set $\Delta\rho_k^{(i)} = \frac{4.8}{3}$ cm and $\Delta\phi_k^{(i)} = \frac{0.47}{3}$.
5. *SC-5*: Set $\Delta\rho_k^{(i)}$ randomly taken from $U(3.6, 4.8)$, in cm, and $\Delta\phi_k^{(i)}$ from $U(0.1173^\circ, 0.47^\circ)$. Set $N_{\text{opt}}^{(i)}$ individually to each sensor.
6. *SC-6*: Consider SC-5 for $Q \leftarrow p^2 Q_k$ with $p = 0.1$.

In terms of the absolute estimation errors, the results are sketched in Fig. 6.7 along the coordinate x and in Fig. 6.8 along y and one can easily trace the differences. The first point to notice is that the dUFIR filter in general outperforms the dKF in each of the above scenarios. To support this inference, the RMSEs computed by the root square of the sum of the MSEs along coordinates x and y are listed in Table 6.3, where the minimum values are bolded.

Of a particular interest is the case of SC-6 illustrated in Fig. 6.8. While the dUFIR estimate remains here unaltered by errors in Q , the dKF reduces the estimation random errors in specific time intervals, such as $1500 \leq k \leq 2000$. However, in $1300 \leq k \leq 1500$ the bias error produced by dKF grows considerably (Fig. 6.9) that speaks in favor of higher robustness of the dUFIR filter.

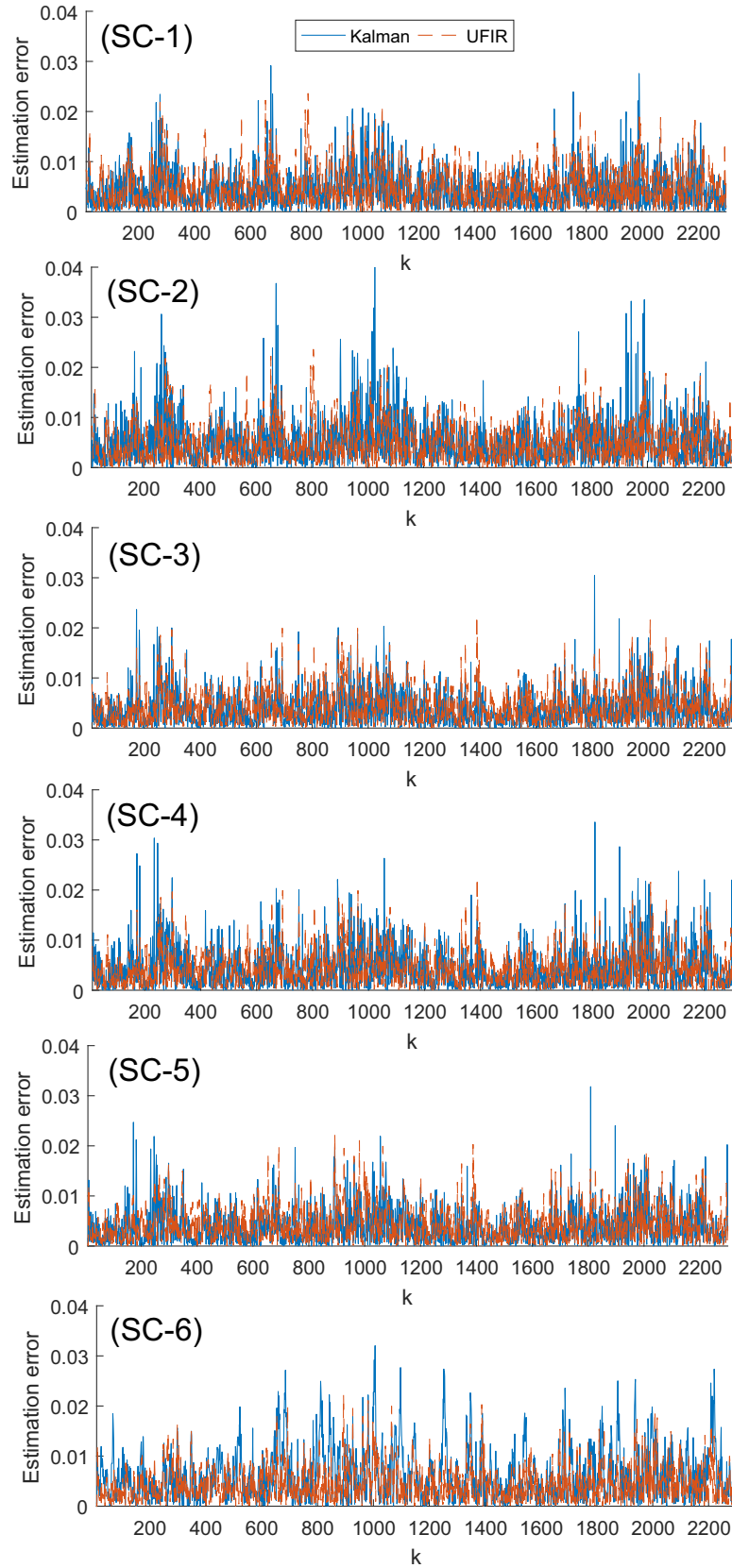


Figure 6.7: Absolute estimation errors along the coordinate x produced by the dKF and dUFIR filter for six scenarios, (SC-1)–(SC-6).

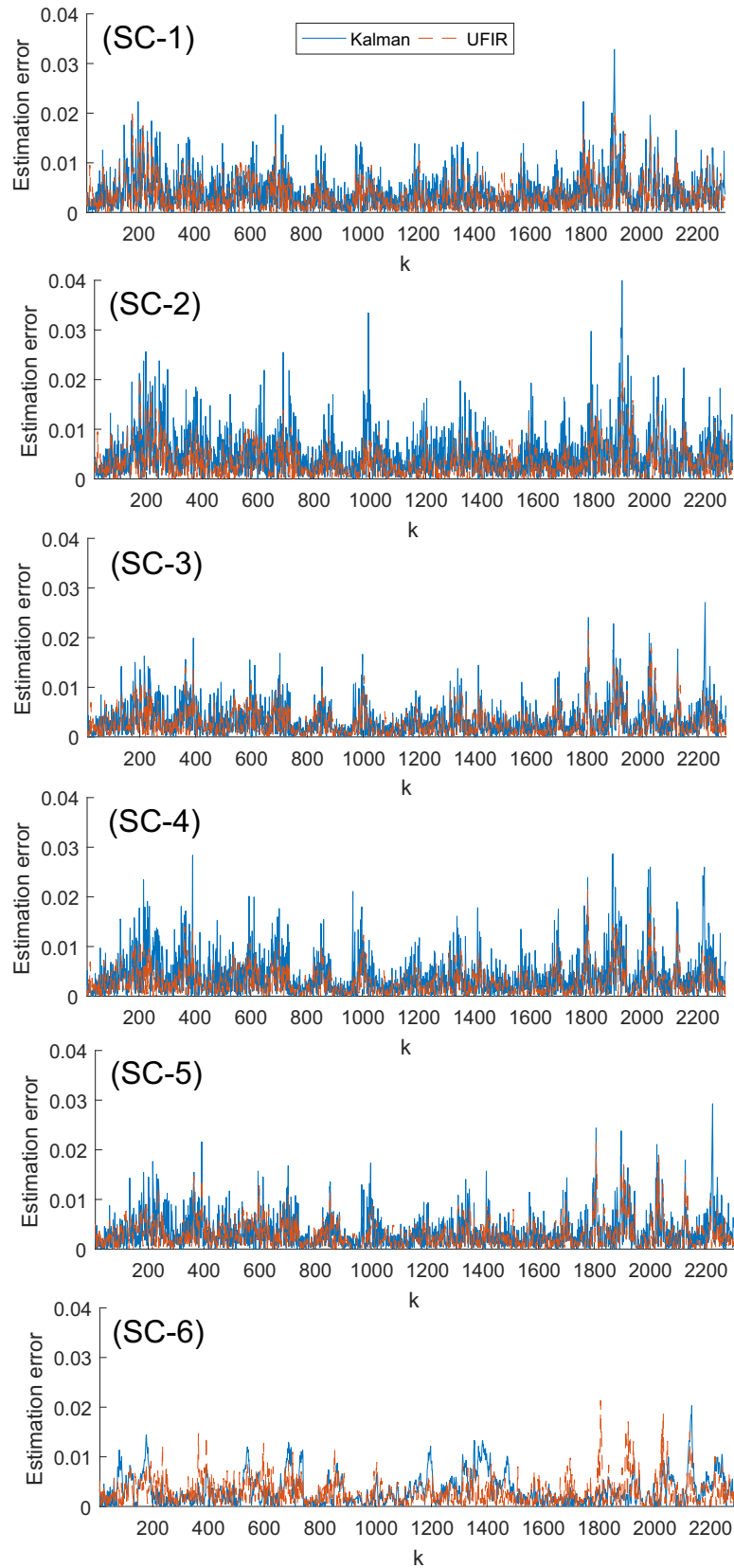


Figure 6.8: Absolute estimation errors along the coordinate y produced by the dKF and dUFIR filter for six scenarios, (SC-1)–(SC-6).

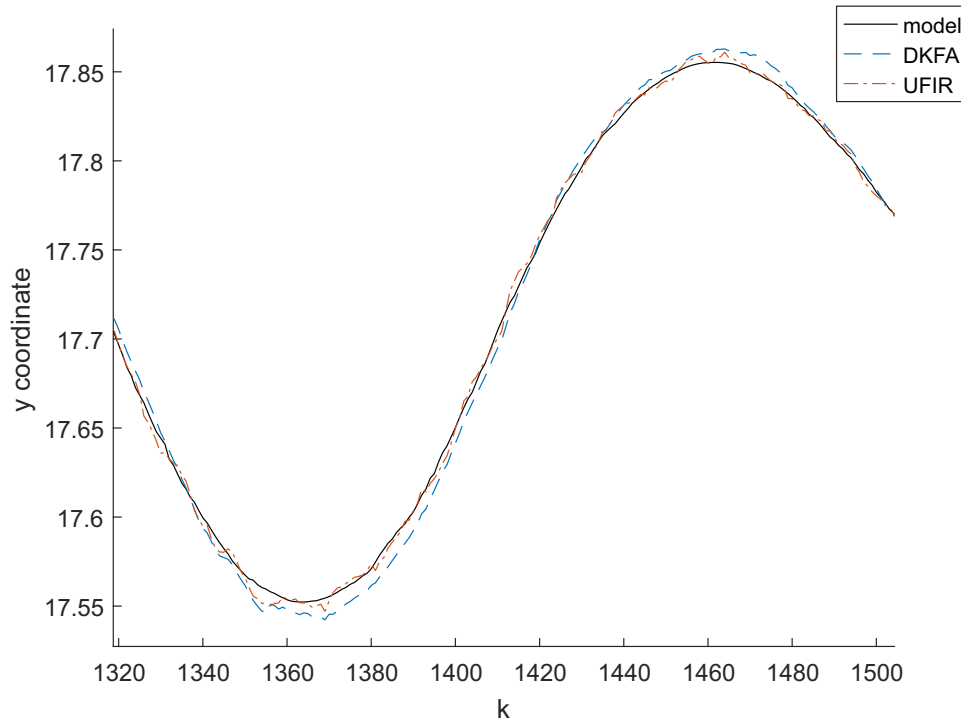


Figure 6.9: Estimates provided by the dKF and dUFIR filter along the coordinate y for scenario SC-6.

6.6 Environmental Temperature Monitoring

In this scenario, we consider temperature measurements taken in 2007 at the Grand-St-Bernard pass at 2400 m. between Switzerland and Italy as part of the Sensorscope project, which aims to develop a large-scale distributed environmental measurement system centered on a wireless sensor network. The temperature measurements were recorded individually by low-cost sensing stations and are available at [140]. In order to test our algorithm a network must be implemented. In Fig. 6.10 we present the topology used for this study, which was the result of considering a link distance of 250 m.

The individual, one-hour average, temperature measurements are sketched in Fig. 6.11 and Fig. 6.12. We observe a similar behavior on all the stations, however for unknown and unpredicted reasons, some sensors present large gaps of information and a very unstable performance.

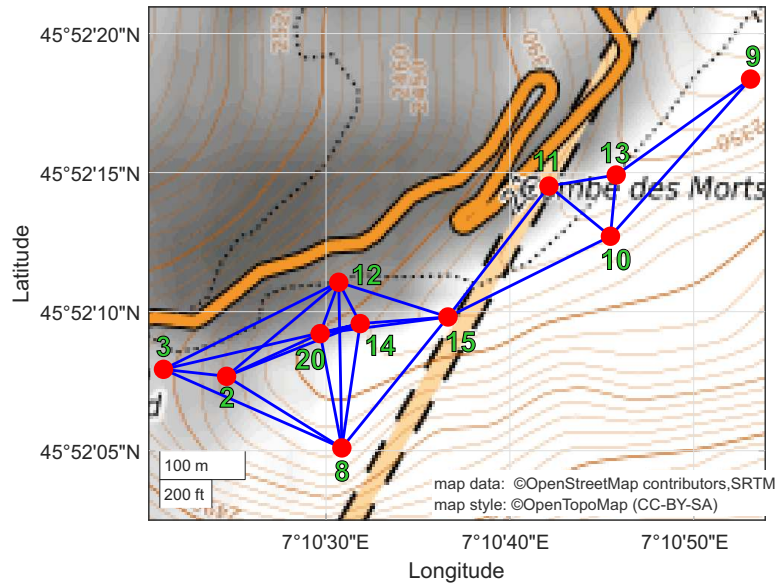


Figure 6.10: Simulated network connections of the sensing stations, given their real locations.

We defined the following state-space discrete dynamics

$$A = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \quad H^{(i)} = \begin{bmatrix} 1 & 0 \end{bmatrix},$$

with $B = I$. An approximation of the individual standard deviation $\sigma_v^{(i)}$, was obtained from the dataset. We also defined $N = 30$ as it was the smaller value that maximized the consensus of estimates.

The results of implementing Algorithm 8 for three sensing stations are shown in Fig. 6.13. We observe a noise reduction in all the stations and a filling of large gaps for station 2 (Fig. 6.13 b)) and 9 (Fig. 6.13 c)).

A key difference between stations 2 and 3 is observable in $540 < k < 780$. While in station 2 measurements are completely lost, on station 9 a false measurement of -1°C is recorded. The algorithm, as it is, cannot discern if the measurement is valid. However, due to the distributive nature of the developed filter, the estimate shows great performance even under such circumstances.

In Fig. 6.14 a), we show the estimates of all the stations while in Fig. 6.14 b) the variance of the estimates is taken as an indication of the disagreement between nodes. It can

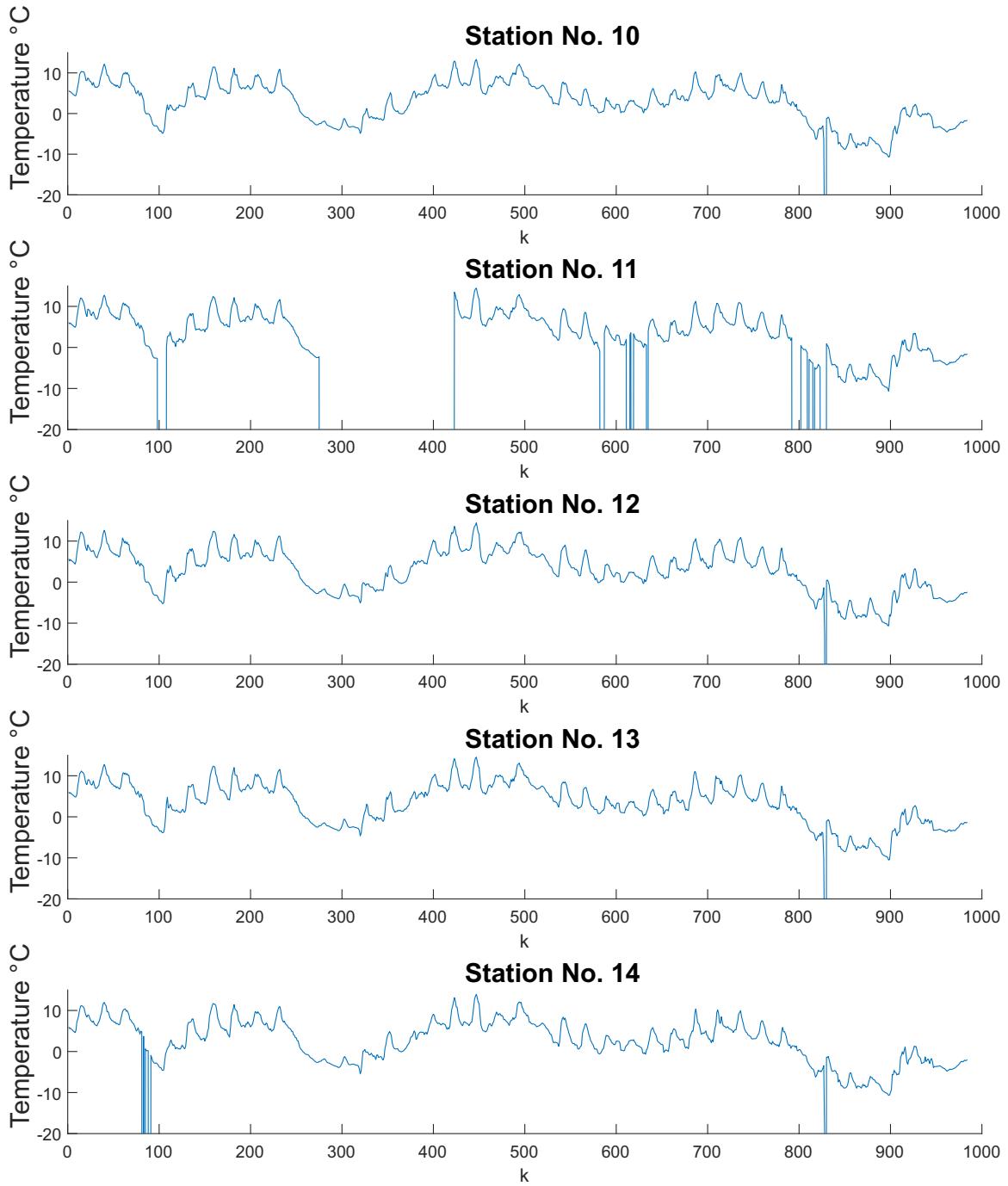


Figure 6.11: Temperature measurements of the eleven sensing stations.

be seen that when valid measurements are taken, the estimates present less disagreement.

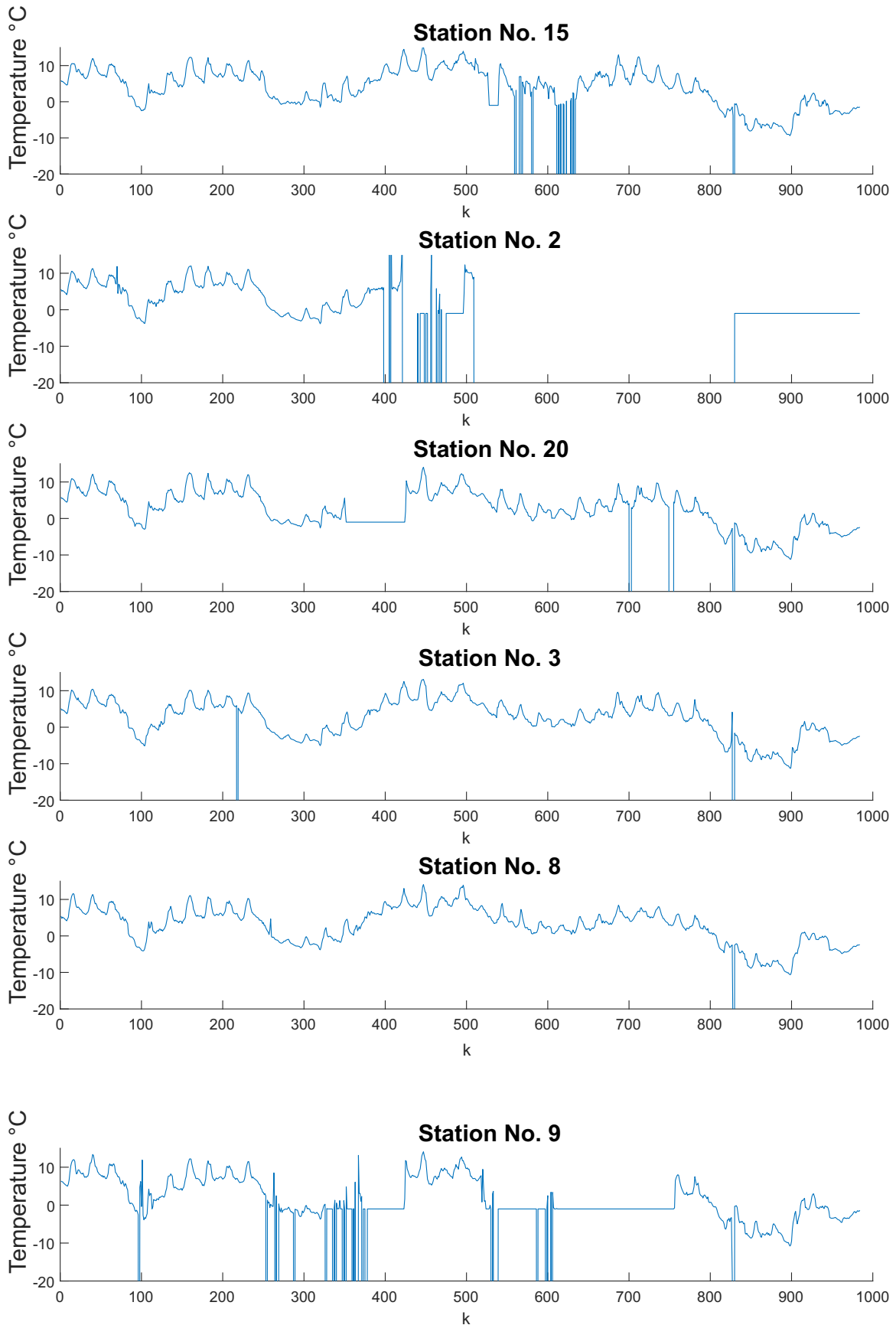


Figure 6.12: Temperature measurements of the eleven sensing stations.

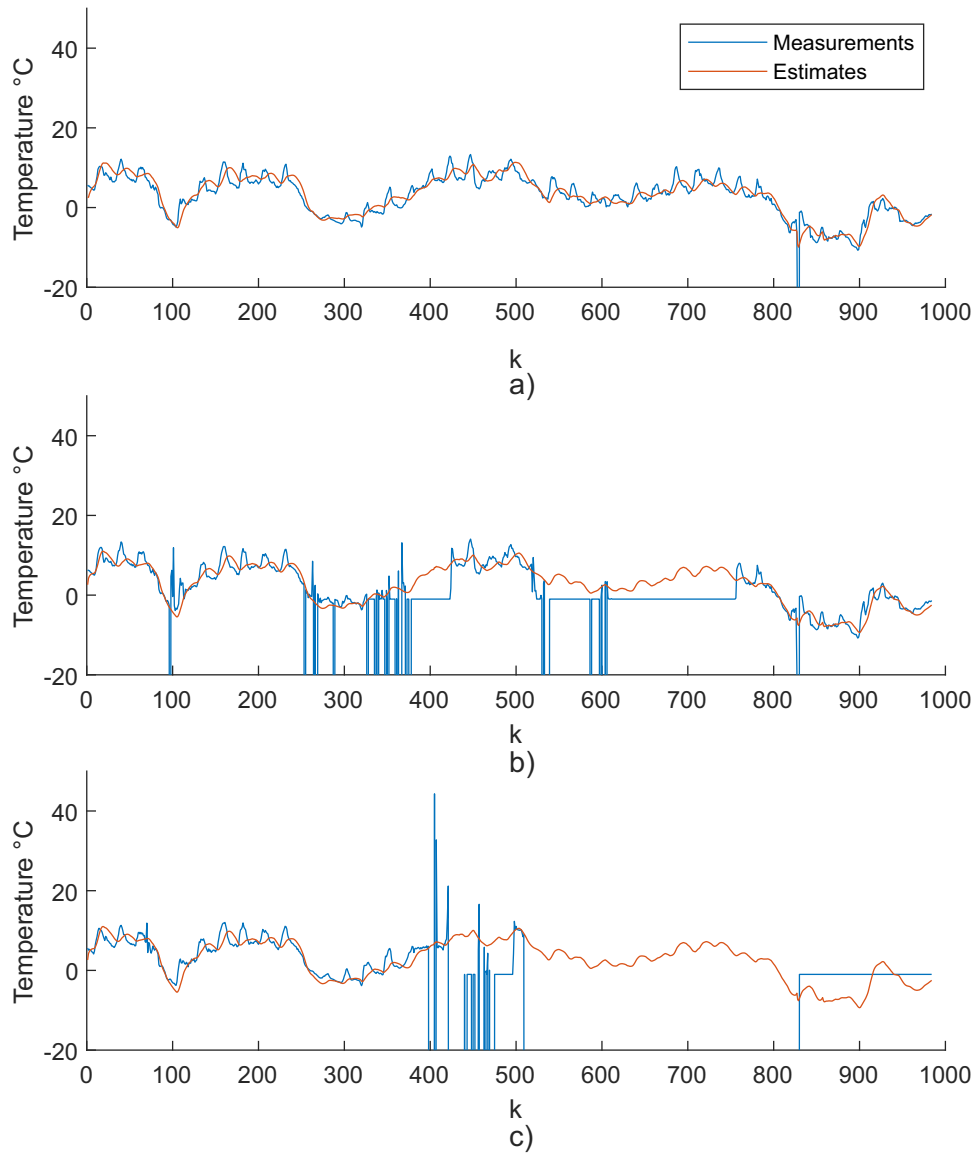


Figure 6.13: Temperature measurements and estimates of: a) station 10, b) station 9 and c) station 2.

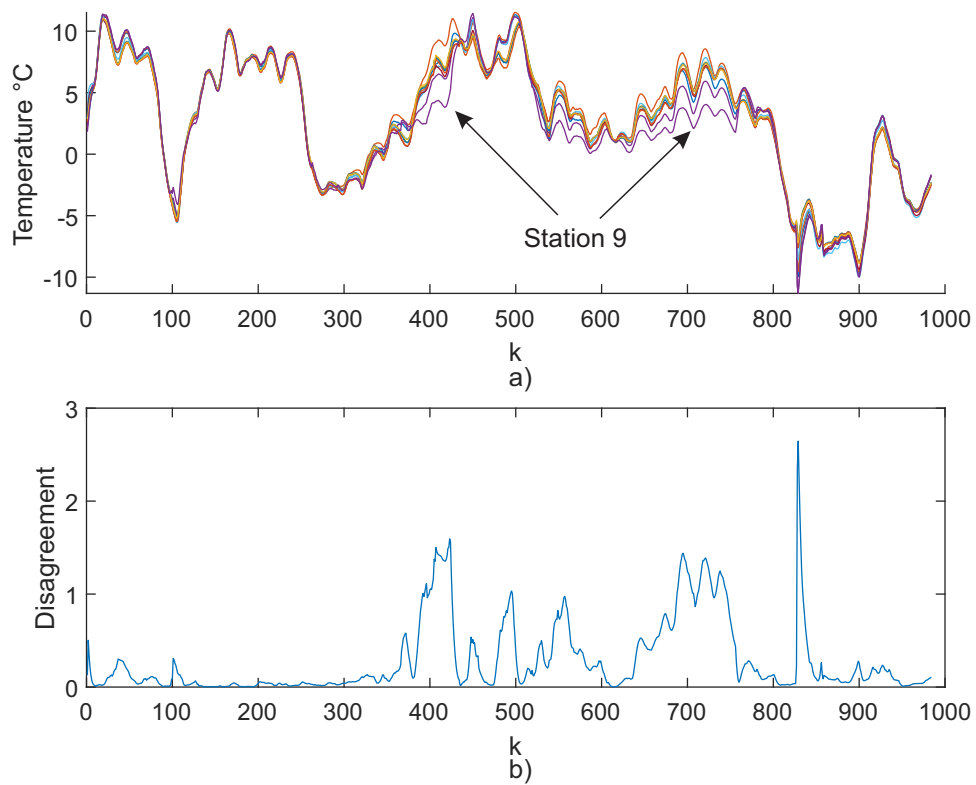


Figure 6.14: a) Temperature estimates of all stations and b) disagreement between estimates.

Chapter 7

Comparison with KF, H_∞ and Generalizations

7.1 Conclusions

The algorithms developed in this work were motivated by a lacking in robustness of one of the most common algorithms for sensor fusion, the KF. Although the KF is easy to implement and its estimations are optimal in the RMSE sense, the lack of robustness against unpredicted errors and unknown noise statistics renders the KF unreliable for WSNs applications.

The main reason behind the lack of robustness of KF is because its IIR structure, because the past errors keep affecting present estimates. It has been demonstrated that better robustness is attainable through FIR structures. However, due to limited memory capacity of the target devices and the need of earlier FIR algorithms to know the correct statistics of noise, the FIR algorithms were not a plausible solution.

Nowadays, the computing capacity of smart devices and the development of Unbiased Finite Impulse Response Filters, makes the FIR algorithms a feasible alternative to KF. However, before this work, the UFIR filter technology was not developed to address the unique requirements of Wireless Sensor Networks.

In this work, we have developed UFIR filters that performs sensor fusion to estimate the states of interest in a distributed manner. Next, we will present several conclusions for

each of the designs presented in this work.

Conclusions: Design of UFIR Filter for Smart Sensors

The iterative UFIR filtering algorithm developed and practically tested in Chapter 3 has demonstrated several properties, which are essential for smart sensing. The UFIR algorithm is blind on a horizon of N points: unlike the KF, it does not require the noise statistics, the initial error statistics, and the initial values. It also has better robustness than the KF. As a special feature, we stress on high accuracy of predictive UFIR filtering over missing data. In fact, practical applications to measurements of CO concentration and temperature with missing data gave impressive results. Even with dozens of lost data points, the UFIR algorithm efficiently bridged the gaps of uncertainties and produced quite accurate estimates.

An important remark needs to be done. In the test performed in chapter 3, implementing the KF with the data at hand was not possible due to missing information about the noise process. This poses an excellent example of why KF cannot always be implemented in real -life applications. Also, like the KF, the UFIR algorithm is a universal estimator. It can be used in smart sensors to solve diverse problems associated with state estimation and filtering.

Once the UFIR filter, in its recursive form, was proven to be suitable for smart sensors, we developed the consensus filters based on the ones developed for KF.

Conclusions: Design of UFIR Filter with Consensus on Measurements for WSNs

In Chapter 4 we developed the micro-UFIR filter for the average consensus on measurements. This filter has demonstrated better robustness than the micro-KF when the WSN is operated under disturbances in not well specified noise environments. The algorithm performs a previous low-pass filter on the measurements, as well as a high-pass filter on the error covariance of the KF. Since UFIR does not need the error covariance, a single low-pass filter applied to the measurements is enough. Operatively speaking, the low-pass

filter is an average of the neighbors measurements. The implementation of the such filter considerably reduces the computing time of UFIR. Better performance of the micro-UFIR filter was shown analytically and supported by simulations of a vehicle traveling circularly on a ground space covered by the WSN. We found that, an increment on the number of links reduces the estimate errors for both, KF and UFIR filter, however, in the latter better performance was obtained with fewer sensors. In order to introduce a new challenging technique, we restricted ourselves only with a homogeneous media and single quantity.

As previously discussed, in [1] the consensus on estimates is an improvement regarding the number of consensus steps. In this work, we followed the same line of thought and developed the distributed UFIR filter with consensus on estimates.

Conclusions: Design of UFIR Filter With Consensus on Estimates for WSNs

The dUFIR filter designed in Chapter 5 for WSNs with consensus on estimates has demonstrated higher accuracy and robustness than the dKF and dH_∞ filter under real world WSN scenarios. The goal was reached by minimizing the MSE in the dUFIR structure with an optimal consensus factor λ_k^{opt} and providing recursions for iterative computation. Although all filters improve the performance with an increase in the number of nodes, the same improvement was obtained in the dUFIR filter with a smaller number of nodes. Simulations and an experimental example of target tracking have revealed that the dUFIR approach is more favorable for distributed WSNs with consensus on estimates, especially under diverse uncertainties. Errors in the dUFIR filter were minimized by finding a the consensus factor λ_k^{opt} , which is optimal in the RMSE sense.

Although the calculation of λ_k^{opt} requires the knowledge of the measurement noise covariance, in real life applications this statistics are relatively easy to approximate through sensor calibration. We also demonstrated by simulating an application where the noise statistics are time-dependent, that even with an approximation of the real value of R the performance of dUFIR filter was superior to that of dKF and dH_∞ .

Conclusions: Design of UFIR Filter With Consensus on Estimates for unstable WSNs

As was discussed in Chapter 3, the predictive capacity of the UFIR filter allows the filter to overcome missing measurements by filling the gaps with predictions. This behavior increases the robustness of the filter, making the algorithm ready for real life operation, where due to a very large number of factors, many missing data will be present. The advantages of a predictive algorithm were exploited in this work by considering real life data of a very unstable network.

In Chapter 6, the problem of object tracking over distributed WSNs with consensus on estimates and missing data has been solved by using the designed dUFIR filter. Better performance of the dUFIR filter-based localization system has been proven with respect to known ground truth through simulations for measurements with missing data and referring to real sensor specifications. Extensive experimental investigations have shown that the dUFIR filter produces smaller errors than the dKF under uncertainties of the noise statistics and model errors. It was also verified that the dUFIR filter allows reaching a better consensus in estimates than the dKF in terms of errors in individual estimates. Another noticeable advantage of the dUFIR filter, which was observed in simulations, is that it requires a smaller number of nodes to achieve the same performance as in the dKF.

The dUFIR algorithm is limited to linear dynamic processes with an approximate knowledge of the sensor noise statistics, and the first N valid measurements. If this requirements are met, the promising results of the filter make it possible to be implemented in applications of indoor tracking where GPS technology is unavailable. For example, in the tracking of mobile robotic platforms in modern industries, indoor tracking of elderly people, tracking of robotics assistants, etc.

The predictive dUFIR filter was also tested with data from a real-life experiment, with large amounts of missing data and where information about system and measurement noise was unavailable. For these reasons, KF could not be implemented. Regarding dUFIR, by not requiring information about system noise the only issue was the lack of measurement noise statistics. In order to overcome this issue, a simple statistical analysis on the measurements was perform, which gave us an approximation of the variance for each sensor.

The distributed Unbiased Finite Impulse Response filter showed great performance even with a large number of missing data. This is the result of implementing both, the prediction stage and the the consensus of estimates. The algorithm is capable of offering estimates even when several hundreds of missing measurements, far larger than the predefined horizon, which is an improvement of the filter discussed in Chapter 3, where after N missing values, the filter started to follow the model, rather than the process. The robustness against missing data is due to the contribution of the neighboring filters. The consensus filter fuses estimations and predictions alike.

7.2 Discussion

Wireless sensor networking is a very vast area of research where many disciplines come together in order to successfully implement such technology in real engineering problems. As far as we are concerned, no attempts were made to develop UFIR filters with average consensus to address the issue of the lack of robustness of IIR filters in WSNs, therefore, in this work we focused only on the estimation problem by ensuring the consensus of the network through the development of unbiased finite impulse response filters with consensus on measurements and estimates, with and without missing data, proving that less estimation error is produced by the UFIR filters when the conditions for optimality of KF are not met.

We also observed that, when properly tuned, H_∞ filter may produce more robust estimates than UFIR filters, however the latter remains as a more viable solution for real life applications. A recursive UFIR filter was also developed in order to minimize the computational burden of the microprocessors, so the filters may be implemented in smart sensors. In applications where there is absolutely no information about the model noise, implementing KF is non-viable. However, UFIR filters with consensus on estimates not only is a viable solution, but produces accurate estimations. An interesting result of this work is the feasibility to obtain adequate estimates for applications where no noise information is available and even more, where a large amount of missing or corrupted data is present.

It is an important remark that, if all the conditions for optimality are met, KF will produce even better results. Moreover, even recursive UFIR filters are fast, it will never be as fast as KF. This is to be considered when implementing the filter in a real WSN.

7.3 Future Work

From this work, the articles [21, 38, 141, 142] were published in journals indexed in the Journal Citation Report, along with other conference papers and indexed journals, however, there is still more research to be done.

A consensus on information with UFIR filters is still under development. Once the filter is developed and proven, a thorough comparison between all type of consensus can be performed. Also, it is possible to implement the different types of consensus with other types of filters, such as particle and H_∞ filters. An important issue is to address the delay in the data or even the duplication of information due to multi-trajectories and of course, the actual implementation in real low cost sensors.

In all the experiments performed in this work, we were interested in the estimation of the variable of interest. However, for some applications, it is also necessary to detect a particular alteration of the model, for example, in forest fire detection. The algorithms can be modified to address this particular applications.

Bibliography

- [1] R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” in *2007 46th IEEE Conf. on Decision and Control*. IEEE, 2007.
- [2] W. Li, G. Wei, and F. Han, “Consensus-based unscented kalman filter for sensor networks with sensor saturations,” in *2014 International Conference on Mechatronics and Control (ICMC)*. IEEE, jul 2014.
- [3] B. Shen, Z. Wang, and Y. Hung, “Distributed H_∞ consensus filtering in sensor networks with multiple missing measurements: The finite-horizon case,” *Automatica*, vol. 46, no. 10, pp. 1682–1688, 2010.
- [4] S. D. Vito, E. Massera, M. Piga, L. Martinotto, and G. D. Francia, “On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario,” *Sensors and Actuators B: Chemical*, vol. 129, no. 2, pp. 750–757, feb 2008.
- [5] S. D. Vito, “Air quality data set,” jun. 2017. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/Air+Quality>
- [6] D. Hanley, A. B. Faustino, S. D. Zelman, D. A. Degenhardt, and T. Bretl, “MagPIE: A dataset for indoor positioning with magnetic anomalies,” in *2017 Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, sep 2017.
- [7] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.
- [8] S. J. Brian D. O. Anderson, *Optimal Filtering*. Englewood Cliff, NJ: Dover Publications Inc., 2005.

-
- [9] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons Inc, 2006.
- [10] Y. S. Shmaliy, S. Zhao, and C. K. Ahn, “Unbiased finite impulse response filtering: an iterative alternative to Kalman filtering ignoring noise and initial conditions,” *IEEE Control Systems Mag.*, vol. 37, no. 5, pp. 70–89, oct 2017.
- [11] M. S. Mahmoud and Y. Xia, *Networked Filtering and Fusion in Wireless Sensor Networks*. CRC Press, 2014.
- [12] G. Li and J. Shi, “Impact of Wireless Sensor Network Technology on Service Innovation in Supply Chain Management,” in *Service Science and Logistics Informatics*. IGI Global, pp. 65–96.
- [13] G. P. Hancke, V. C. Gungor, and G. P. Hancke, “Guest editorial special section on industrial wireless sensor networks,” *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 762–765, feb 2014.
- [14] D. Larios, J. Barbancho, F. Molina, and C. León, “LIS: localization based on an intelligent distributed fuzzy system applied to a WSN,” *Ad Hoc Networks*, vol. 10, no. 3, pp. 604–622, may 2012.
- [15] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar 2002.
- [16] C. Chen, S. Zhu, X. Guan, and X. Shen, *Wireless Sensor Networks: Distributed Consensus Estimation*. Springer Intern. Publ., 2014.
- [17] B. Rao and H. Durrant-Whyte, “Fully decentralised algorithm for multisensor Kalman filtering,” *IEE Proc. D Control Theory and Applic.*, vol. 138, no. 5, p. 413, 1991.
- [18] D. Wang, L. Lin, and L. Xu, “A study of subdividing hexagon-clustered WSN for power saving: analysis and simulation,” *Ad Hoc Networks*, vol. 9, no. 7, pp. 1302–1311, Sep 2011.

- [19] M. Kohvakka, J. Suhonen, M. Kuorilehto, V. Kaseva, M. Hännikäinen, and T. D. Hämäläinen, “Energy-efficient neighbor discovery protocol for mobile wireless sensor networks,” *Ad Hoc Networks*, vol. 7, no. 1, pp. 24–41, Jan 2009.
- [20] J. Roselin, P. Latha, and S. Benitta, “Maximizing the wireless sensor networks lifetime through energy efficient connected coverage,” *Ad Hoc Networks*, vol. 62, pp. 1–10, Jul 2017.
- [21] M. Vazquez-Olguin, Y. S. Shmaliy, and O. G. Ibarra-Manzano, “Distributed unbiased FIR filtering with average consensus on measurements for WSNs,” *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1440–1447, Jun 2017.
- [22] H. Dong, Z. Wang, and H. Gao, “Distributed filtering for a class of time-varying systems over sensor networks with quantization errors and successive packet dropouts,” *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 3164–3173, Jun 2012.
- [23] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, “Consensus in Ad Hoc WSNs with noisy links—Part I: Distributed estimation of deterministic signals,” *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan 2008.
- [24] E. Nurellari, D. McLernon, and M. Ghogho, “Distributed two-step quantized fusion rules via consensus algorithm for distributed detection in wireless sensor networks,” *IEEE Trans. Signal Informat. Process. over Networks*, vol. 2, no. 3, pp. 321–335, Sep. 2016.
- [25] R. Olfati-Saber and R. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, sep 2004.
- [26] R. Olfati-Saber, “Distributed Kalman filter with embedded consensus filters,” in *Proc. 44th IEEE Conf. Decision and Control*. IEEE, 2005.
- [27] —, “Kalman-consensus filter : Optimality, stability, and performance,” in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, dec 2009.

- [28] G. Battistelli, L. Chisci, S. Morrocchi, and F. Papi, “An information-theoretic approach to distributed state estimation,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 12 477–12 482, jan 2011.
- [29] G. Battistelli and L. Chisci, “Kullback–leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability,” *Automatica*, vol. 50, no. 3, pp. 707–718, mar 2014.
- [30] X. Bai, Z. Wang, L. Zou, and F. E. Alsaadi, “Collaborative fusion estimation over wireless sensor networks for monitoring CO 2 concentration in a greenhouse,” *Information Fusion*, vol. 42, pp. 119–126, Jul 2018.
- [31] S. Zhao, Y. S. Shmaliy, and C. K. Ahn, “Bias-constrained optimal fusion filtering for decentralized WSN with correlated noise sources,” *IEEE Trans. Signal Informat. Process. over Networks*, vol. 4, no. 4, pp. 727–735, Dec 2018.
- [32] S. Zhao, Y. S. Shmaliy, B. Huang, and F. Liu, “Minimum variance unbiased FIR filter for discrete time-variant systems,” *Automatica*, vol. 53, pp. 355–361, mar 2015.
- [33] W. H. Kwon, P. S. Kim, and P. Park, “A receding horizon kalman FIR filter for linear continuous-time systems,” *IEEE Transactions on Automatic Control*, vol. 44, no. 11, pp. 2115–2120, 1999.
- [34] A. H. Jazwinski, *Stochastic Processes and Filtering Theory (Mathematics in Science and Engineering)*. Academic Press, 1970.
- [35] Y. Shmaliy, “A simple optimally unbiased MA filter for timekeeping,” *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 49, no. 6, pp. 789–797, jun 2002.
- [36] —, “An unbiased FIR filter for TIE model of a local clock in applications to GPS-based timekeeping,” *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 53, no. 5, pp. 862–870, may 2006.

- [37] Y. S. Shmaliy, “An iterative kalman-like algorithm ignoring noise and initial conditions,” *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2465–2473, jun 2011.
- [38] M. Vazquez-Olguin, Y. S. Shmaliy, C. K. Ahn, and O. G. Ibarra-Manzano, “Blind robust estimation with missing data for smart sensors using UFIR filtering,” *IEEE Sensors Journal*, vol. 17, no. 6, pp. 1819–1827, mar 2017.
- [39] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, “Multisensor data fusion: A review of the state-of-the-art,” *Information Fusion*, vol. 14, no. 1, pp. 28–44, jan 2013.
- [40] W. Li, Z. Wang, G. Wei, L. Ma, J. Hu, and D. Ding, “A survey on multisensor fusion and consensus filtering for sensor networks,” *Discrete Dynamics in Nature and Society*, vol. 2015, pp. 1–12, 2015.
- [41] R. Olfati-Saber and J. Shamma, “Consensus filters for sensor networks and distributed sensor fusion,” in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE.
- [42] J. J. Pomarico-Franquiz and Y. S. Shmaliy, “Accurate self-localization in RFID tag information grids using FIR filtering,” *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1317–1326, May 2014.
- [43] Y. S. Shmaliy, S. Khan, and S. Zhao, “Ultimate iterative UFIR filtering algorithm,” *Measurement*, vol. 92, pp. 236–242, Oct 2016.
- [44] D. Y. Kim and M. Jeon, “Robust Distributed Kalman Filter for Wireless Sensor Networks with Uncertain Communication Channels,” *Mathematical Problems in Engineering*, vol. 2012, pp. 1–12, 2012.
- [45] H. Yu, Y. Zhuang, and W. Wang, “Distributed H_∞ filtering with consensus in sensor networks: a two-dimensional system-based approach,” *International Journal of Systems Science*, vol. 42, no. 9, pp. 1543–1557, sep 2011.

- [46] A. H. Jazwinski, *Stochastic Processes and Filtering Theory (Dover Books on Electrical Engineering)*. Dover Publications, 2007.
- [47] S. Zhao, Y. S. Shmaliy, and F. Liu, “Fast computation of discrete optimal FIR estimates in white gaussian noise,” *IEEE Signal Process. Lett.*, vol. 22, no. 6, pp. 718–722, jun 2015.
- [48] M. Farina, G. Ferrari-Trecate, and R. Scattolini, “Distributed moving horizon estimation for linear constrained systems,” *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2462–2475, nov 2010.
- [49] D. Ding, Z. Wang, H. Dong, and H. Shu, “Distributed H_∞ state estimation with stochastic parameters and nonlinearities through sensor networks: The finite-horizon case,” *Automatica*, vol. 48, no. 8, pp. 1575–1585, aug 2012.
- [50] W. Shang, Y. Kang, H. Xi, Y. Xia, and Y.-B. Zhao, “Distributed H_∞ consensus filtering with sensor networks: a finite horizon solution,” *IMA Journal of Mathematical Control and Information*, vol. 31, no. 1, pp. 33–49, mar 2013.
- [51] S. H. Han, W. H. Kwon, and P. S. Kim, “Quasi-deadbeat minimax filters for deterministic state~space models,” *IEEE Transactions on Automatic Control*, vol. 47, no. 11, pp. 1904–1908, nov 2002.
- [52] Y. S. Shmaliy and O. Ibarra-Manzano, “Noise power gain for discrete-time FIR estimators,” *IEEE Signal Process. Lett.*, vol. 18, no. 4, pp. 207–210, Apr 2011.
- [53] Y. S. Shmaliy and L. Arceo-Miquel, “Efficient predictive estimator for holdover in GPS-based clock synchronization,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 55, no. 10, pp. 2131–2139, October 2008.
- [54] J. B. Fu, J. Sun, , and S. Lu, “Maneuvering target tracking with improved unbiased FIR filter,” in *2014 International Radar Conference*, Oct 2014, pp. 1–5.
- [55] S. Zhao, Y. S. Shmaliy, and F. Liu, “Fast Kalman-Like Optimal Unbiased FIR Filtering With Applications,” *IEEE Transactions on Signal Processing*, vol. 64, no. 9, pp. 2284–2297, may 2016.

- [56] C. K. Ahn, P. Shi, and M. V. Basin, "Deadbeat dissipative fir filtering," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 8, pp. 1210–1221, Aug 2016.
- [57] C. K. Ahn, Y. S. Shmaliy, P. Shi, and Y. Zhao, "Receding-Horizon l_2-l_∞ FIR Filter With Embedded Deadbeat Property," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 2, pp. 211–215, Feb 2017.
- [58] J. M. Pak, C. K. Ahn, Y. S. Shmaliy, and M. T. Lim, "Improving Reliability of Particle Filter-Based Localization in Wireless Sensor Networks via Hybrid Particle/FIR Filtering," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 5, pp. 1089–1098, Oct 2015.
- [59] J. M. Pak, C. K. Ahn, Y. S. Shmaliy, P. Shi, and M. T. Lim, "Switching Extensible FIR Filter Bank for Adaptive Horizon State Estimation With Application," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 1052–1058, May 2016.
- [60] J. M. Giachino, "Smart sensors," *Sensors and actuators*, vol. 10, no. 3-4, pp. 239–248, 1986.
- [61] "Ieee standard for a smart transducer interface for sensors and actuators - transducer to microprocessor communication protocols and transducer electronic data sheet (teds) formats," *IEEE Std 1451.2-1997*, pp. i–, 1998.
- [62] E. Song and K. Lee, "Understanding IEEE 1451-Networked smart transducer interface standard - What is a smart transducer?" *IEEE Instrumentation & Measurement Magazine*, vol. 11, no. 2, pp. 11–17, apr 2008.
- [63] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*. Piscataway, NJ: Wiley-Blackwell, 2009.
- [64] S. Chalasani and J. M. Conrad, "A survey of energy harvesting sources for embedded systems," in *IEEE SoutheastCon 2008*. IEEE, apr 2008.

- [65] C. P. Waltenegus Dargie, *Fundamentals of Wireless Sensor Networks*. John Wiley & Sons, 2010.
- [66] C. Townsend and S. Arms, “Wireless sensor networks,” in *Sensor Technology Handbook*. Elsevier, 2005, pp. 575–589.
- [67] L. Yong-Min, W. Shu-Ci, and N. Xiao-Hong, “The architecture and characteristics of wireless sensor network,” in *2009 International Conference on Computer Technology and Development*. IEEE, 2009.
- [68] R. Saber and R. Murray, “Consensus protocols for networks of dynamic agents,” in *Proceedings of the 2003 American Control Conference, 2003*. IEEE.
- [69] “Information consensus in multivehicle cooperative control,” *IEEE Control Systems*, vol. 27, no. 2, pp. 71–82, apr 2007.
- [70] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, jan 2007.
- [71] R. Frank, *Understanding Smart Sensors*. Norwood, MA: Artech House, 2013.
- [72] P. G. Ranky, “Smart sensors,” *Sensor Review*, vol. 22, no. 4, pp. 312–318, dec 2002.
- [73] D. R. H. Phillip E. Allen, *CMOS Analog Circuit Design*. OXFORD UNIV PR, 2011.
- [74] R. Landry, “Mixed signal soc applications,” Design and Reuse.
- [75] E. T. Powner and F. Yalcinkaya, “From basic sensors to intelligent sensors: Definitions and examples,” *Sensor Review*, vol. 15, no. 4, pp. 19–22, dec 1995.
- [76] M. Sveda and R. Vrba, “Integrated smart sensor networking framework for sensor-based appliances,” *IEEE Sensors Journal*, vol. 3, no. 5, pp. 579–586, oct 2003.

- [77] M. Mailand, S. Getzlaff, D. Breitmeyer, R. Richter, U. Meyer, E. Boehme, and A. Krasselt, “A flexible, programmable sensor signal conditioning IC for high-precision smart sensors,” in *2011 Semiconductor Conference Dresden*. IEEE, sep 2011.
- [78] D. C. Swanson, *Signal Processing for Intelligent Sensor Systems with MATLAB*. CRC Press, 2017.
- [79] N. V. Kirianaki, *Data Acquisition and Signal Processing for Smart Sensors*. Wiley-Blackwell, 2002.
- [80] L. G. Feng Zhao, *Wireless Sensor Networks: An Information Processing Approach*. MORGAN KAUFMANN PUBL INC, 2004.
- [81] H. Taghirad and P. Belanger, “Torque ripple and misalignment torque compensation for the built-in torque sensor of harmonic drive systems,” *IEEE Transactions on Instrumentation and Measurement*, vol. 47, no. 1, pp. 309–315, 1998.
- [82] S.-H. Won, F. Golnaraghi, and W. Melek, “A fastening tool tracking system using an IMU and a position sensor with kalman filters and a fuzzy expert system,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 5, pp. 1782–1792, may 2009.
- [83] R. Blind, S. Uhlich, B. Yang, and F. Allgower, “Robustification and optimization of a kalman filter with measurement loss using linear precoding,” in *2009 American Control Conference*. IEEE, 2009.
- [84] K. H. Eom, S. J. Lee, Y. S. Kyung, C. W. Lee, M. C. Kim, and K. K. Jung, “Improved kalman filter method for measurement noise reduction in multi sensor RFID systems,” *Sensors*, vol. 11, no. 11, pp. 10 266–10 282, oct 2011.
- [85] M. Nourian, A. S. Leong, S. Dey, and D. E. Quevedo, “An optimal transmission strategy for kalman filtering over packet dropping links with imperfect acknowledgements,” *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 259–271, sep 2014.

- [86] R. Kannan, "Orientation estimation based on LKF using differential state equation," *IEEE Sensors Journal*, vol. 15, no. 11, pp. 6156–6163, nov 2015.
- [87] Y. Zhuang, Y. Li, L. Qi, H. Lan, J. Yang, and N. El-Sheimy, "A two-filter integration of MEMS sensors and WiFi fingerprinting for indoor positioning," *IEEE Sensors Journal*, vol. 16, no. 13, pp. 5125–5126, jul 2016.
- [88] H. Benítez Pérez, J. L. Ortega Arjona, and G. R. Latif Shabgahi, "Definition and empirical evaluation of voters for redundant smart sensor systems," *Computación y Sistemas*, vol. 11, no. 1, pp. 39–60, 2007.
- [89] R. Doraiswami and J. Jiang, "An intelligent sensor to monitor power system stability, performance and diagnose failures," *IEEE Transactions on Power Systems*, vol. 5, no. 4, pp. 1432–1438, 1990.
- [90] M. P. Schoen, "Dynamic compensation of intelligent sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 5, pp. 1992–2001, oct 2007.
- [91] J. Patra, A. Kot, and G. Panda, "An intelligent pressure sensor using neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 4, pp. 829–834, 2000.
- [92] N. N. Charniya and S. V. Dudul, "Intelligent sensor system for discrimination of material type using neural networks," *Applied Soft Computing*, vol. 12, no. 1, pp. 543–552, jan 2012.
- [93] Y. S. Shmaliy and O. Ibarra-Manzano, "Time-variant linear optimal finite impulse response estimator for discrete state-space models," *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 2, pp. 95–104, sep 2011.
- [94] H. Rezaei, R. M. Esfanjani, and M. H. Sedaaghi, "Improved robust finite-horizon kalman filtering for uncertain networked time-varying systems," *Information Sciences*, vol. 293, pp. 263–274, feb 2015.

- [95] Y. S. Shmaliy, “Linear optimal FIR estimation of discrete time-invariant state-space models,” *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3086–3096, jun 2010.
- [96] F. Auger, M. Hilairet, J. M. Guerrero, E. Monmasson, T. Orlowska-Kowalska, and S. Katsura, “Industrial applications of the kalman filter: A review,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5458–5471, dec 2013.
- [97] Y. S. Shmaliy and D. Simon, “Iterative unbiased FIR state estimation: a review of algorithms,” *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, may 2013.
- [98] F. Ramirez-Echeverria, A. Sarr, and Y. S. Shmaliy, “Optimal memory for discrete-time FIR filters in state-space,” *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 557–561, feb 2014.
- [99] P.-S. Kim and M.-E. Lee, “A new FIR filter for state estimation and its application,” *Journal of Computer Science and Technology*, vol. 22, no. 5, pp. 779–784, sep 2007.
- [100] Y. S. Shmaliy, S. H. Khan, S. Zhao, and O. Ibarra-Manzano, “General unbiased FIR filter with applications to GPS-based steering of oscillator frequency,” *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 1141–1148, may 2017.
- [101] V. Sima, *Algorithms for Linear-Quadratic Optimization (Chapman & Hall/CRC Pure and Applied Mathematics)*. Chapman and Hall/CRC, 1996.
- [102] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall, 2000.
- [103] C. Ahn, S. Han, and W. Kwon, “Infinite FIR filters for linear continuous-time state-space systems,” *IEEE Signal Processing Letters*, vol. 13, no. 9, pp. 557–560, sep 2006.
- [104] C. K. Ahn, “Strictly passive FIR filtering for state-space models with external disturbance,” *AEU - International Journal of Electronics and Communications*, vol. 66, no. 11, pp. 944–948, nov 2012.

-
- [105] J. M. Pak, C. K. Ahn, M. T. Lim, and M. K. Song, “Horizon group shift FIR filter: Alternative nonlinear filter using finite recent measurements,” *Measurement*, vol. 57, pp. 33–45, nov 2014.
- [106] N. de Nevers, *Air Pollution Control Engineering*. Waveland Pr Inc, 2010.
- [107] T. Godish, W. T. Davis, and J. S. Fu, *Air Quality, Fifth Edition*. ROUTLEDGE, 2014.
- [108] S. K. Yung and D. W. Clarke, “Local sensor validation,” *Measurement and Control*, vol. 22, no. 5, pp. 132–141, jun 1989.
- [109] R. Budampati and S. Kolavennu, *Industrial Wireless Sensor Networks: Monitoring, Control and Automation*. WOODHEAD PUB, 2015.
- [110] V. Çağrı Güngör and G. P. Hancke, *Industrial Wireless Sensor Networks: Applications, Protocols, and Standards (Industrial Electronics)*. CRC Press, 2017.
- [111] Y.-R. Tsai and C.-J. Chang, “Cooperative information aggregation for distributed estimation in wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3876–3888, aug 2011.
- [112] S. Das and J. M. F. Moura, “Distributed Kalman filtering with dynamic observations consensus,” *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 1458–1473, Sep. 2015.
- [113] D. Scherber and H. Papadopoulos, “Distributed computation of averages over ad hoc networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 776–787, apr 2005.
- [114] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005*. IEEE.
- [115] I. D. Schizas, G. B. Giannakis, S. I. Roumeliotis, and A. Ribeiro, “Consensus in ad hoc WSNs with noisy links—part II: Distributed estimation and smoothing of

- random signals,” *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1650–1666, apr 2008.
- [116] S. Kar and J. Moura, “Sensor networks with random links: Topology design for distributed consensus,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3315–3326, jul 2008.
- [117] —, “Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise,” *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, jan 2009.
- [118] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, “Distributed kalman filtering based on consensus strategies,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 622–633, may 2008.
- [119] D. J. Simon, “From here to infinity,” *Embedded Systems Programming*, vol. 14, no. 11, p. 20, 2001.
- [120] W. H. Kwon and S. H. Han, *Receding horizon control: model predictive control for state models*. Springer Science & Business Media, 2006.
- [121] B. P. Gibbs, *Advanced Kalman Filtering, Least-Squares and Modeling*. Wiley-Blackwell, 2011.
- [122] N. Rahbari-Asr and M.-Y. Chow, “Cooperative distributed demand management for community charging of PHEV/PEVs based on KKT conditions and consensus networks,” *IEEE Trans. Ind. Informat.*, vol. 10, no. 3, pp. 1907–1916, Aug. 2014.
- [123] W. Yu, L. Zhou, X. Yu, J. Lu, and R. Lu, “Consensus in multi-agent systems with second-order dynamics and sampled data,” *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2137–2146, Nov. 2013.
- [124] S. S. Stanković, M. S. Stanković, and D. M. Stipanović, “Consensus based overlapping decentralized estimation with missing observations and communication faults,” *Automatica*, vol. 45, no. 6, pp. 1397–1406, jun 2009.

- [125] W. Wang, H. Ma, Y. Wang, and M. Fu, "Performance analysis based on least squares and extended Kalman filter for localization of static target in wireless sensor networks," *Ad Hoc Networks*, vol. 25, pp. 1–15, feb 2015.
- [126] S. R. Jondhale and R. S. Deshpande, "GRNN and KF framework based real time target tracking using PSOC BLE and smartphone," *Ad Hoc Networks*, sep 2018.
- [127] K. Uribe-Murcia, Y. S. Shmaliy, and J. A. Andrade-Lucio, "UFIR filtering for GPS-based tracking over WSNs with delayed and missing data," *Journal of Electrical and Computer Engineering*, vol. 2018, pp. 1–9, may 2018.
- [128] X. Fan, S. Zhao, and Y. S. Shmaiy, "Linear optimal fusion of local unbiased FIR filters," *MATEC Web of Conferences*, vol. 210, p. 05004, 2018.
- [129] X. Fan and F. Liu, "State fusion of decentralized optimal unbiased FIR filters," *J. Electr. Comput. Eng.*, vol. 2018, pp. 1–11, jun 2018.
- [130] M. Vazquez-Olguin, Y. S. Shmaliy, and O. Ibarra-Manzano, "Developing UFIR Filtering with Consensus on Estimates for Distributed Wireless Sensor Networks," *WSEAS Trans. Circuits Syst.*, 2018.
- [131] Y. S. Shmaliy, F. Lehmann, S. Zhao, and C. K. Ahn, "Comparing Robustness of the Kalman, H_∞ , and UFIR Filters," *IEEE Transactions on Signal Processing*, vol. 66, no. 13, pp. 3447–3458, jul 2018.
- [132] A. Kelly, "Precision dilution in mobile robot position estimation," in *Proceedings of Autonomous Systems*, 2003.
- [133] A. Bahr and J. J. Leonard, "Minimizing trilateration errors in the presence of uncertain landmark positions," *3rd European Conference on Mobile Robots (ECMR)*, pp. 48–53. [Online]. Available: <http://infoscience.epfl.ch/record/134044>
- [134] J. Lin, W. Xiao, F. Lewis, and L. Xie, "Energy-efficient distributed adaptive multisensor scheduling for target tracking in wireless sensor networks," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 6, pp. 1886–1896, Jun 2009.

-
- [135] C. Liang, F. Wen, and Z. Wang, "Trust-based distributed Kalman filtering for target tracking under malicious cyber attacks," *Informat. Fusion*, vol. 46, pp. 44–50, Mar 2019.
- [136] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, Aug 2008.
- [137] H. Qian, P. Fu, B. Li, J. Liu, and X. Yuan, "A novel loss recovery and tracking scheme for maneuvering target in hybrid WSNs," *Sensors*, vol. 18, no. 2, p. 341, Jan 2018.
- [138] L. Pan, H. Gao, H. Gao, and Y. Liu, "A spatial correlation based adaptive missing data estimation algorithm in wireless sensor networks," *Int. J. Wireless Informat. Networks*, vol. 21, no. 4, pp. 280–289, Oct 2014.
- [139] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, sep 2004.
- [140] G. Barrenetxea, "Sensorscope data," 2019.
- [141] M. Vazquez-Olguin, Y. S. Shmaliy, and O. Ibarra-Manzano, "Distributed UFIR filtering over WSNs with consensus on estimates," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.
- [142] M. Vazquez-Olguin, Y. S. Shmaliy, O. Ibarra-Manzano, J. Munoz-Minjares, and C. Lastre-Dominguez, "Object tracking over distributed WSNs with consensus on estimates and missing data," *IEEE Access*, vol. 7, pp. 39 448–39 458, 2019.

Appendix A

Derivation of the Consensus Factor

λ_k^{opt}

Consider the error covariance $P_k = E\{\varepsilon_k \varepsilon_k^T\}$ as function of λ_k , to be

$$\begin{aligned}
P_k &= (\bar{D}_{m,k} - \tilde{\Theta}_{m,k} M_{m,k}) Q_{m,k} (\bar{D}_{m,k} - \tilde{\Theta}_{m,k} M_{m,k})^T \\
&\quad + J \lambda_k (\tilde{\Theta}_{m,k} \bar{R}_{m,k} \tilde{\Theta}_{m,k}^T - \Theta_{m,k}^{(i)} \tilde{R}^{(i)T} \tilde{\Theta}_{m,k}^T) \\
&\quad + J [\lambda_k (\tilde{\Theta}_{m,k} \bar{R}_{m,k} \tilde{\Theta}_{m,k}^T - \Theta_{m,k}^{(i)} \tilde{R}^{(i)T} \tilde{\Theta}_{m,k}^T)]^T \\
&\quad + J^2 \lambda_k (\tilde{\Theta}_{m,k} \bar{R}_{m,k} \tilde{\Theta}_{m,k}^T - \Theta_{m,k}^{(i)} \tilde{R}^{(i)} \Theta_{m,k}^{(i)T}) \lambda_k^T \\
&\quad + J^2 \lambda_k (\Theta_{m,k}^{(i)} \tilde{R}^{(i)T} \tilde{\Theta}_{m,k}^T + \Theta_{m,k}^{(i)} \bar{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T}) \lambda_k^T, \tag{A.1}
\end{aligned}$$

where $\tilde{\Theta}_{m,k} = G_k \mathcal{H}_{m,k}^T$ and

$$\begin{aligned}
\bar{R}_{m,k} &= E\{v_{m,k} v_{m,k}^T\} = \text{diag}(R_m \dots R_k), \\
\bar{R}_{m,k}^{(i)} &= E\{v_{m,k}^{(i)} v_{m,k}^{(i)T}\} = \text{diag}(R_m^{(i)} \dots R_k^{(i)}), \\
\tilde{R}_{m,k}^{(i)} &= E\{v_{m,k} v_{m,k}^{(i)T}\} = \text{diag}(\tilde{R}_m^{(i)} \dots \tilde{R}_k^{(i)}).
\end{aligned}$$

We next apply the derivative with respect to λ_k to the trace of (A.1) by using the identities $\frac{\partial}{\partial X} \text{tr}(X^T B X) = B X + B^T X$, and $\frac{\partial}{\partial X} \text{tr}(X A) = A^T$. Putting the derivative to zero yields

$$\begin{aligned}
\lambda_k^{\text{opt}} &= -\frac{1}{J} (\tilde{\Theta}_{m,k} \bar{R}_{m,k} \tilde{\Theta}_{m,k}^T - \tilde{\Theta}_{m,k} \tilde{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T}) \times \\
&\quad (\tilde{\Theta}_{m,k} \bar{R}_{m,k} \tilde{\Theta}_{m,k}^T - \tilde{\Theta}_{m,k} \tilde{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T} - (\tilde{\Theta}_{m,k} \tilde{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T})^T + \Theta_{m,k}^{(i)} \bar{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T}), \tag{A.2}
\end{aligned}$$

and with the identities

$$\begin{aligned}\tilde{\Theta}_{m,k} \tilde{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T} &= (\tilde{\Theta}_{m,k} \tilde{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T})^T, \\ \tilde{\Theta}_{m,k} \tilde{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T} &= G_k G_k^{(i)-1} \Theta_{m,k}^{(i)} \bar{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T},\end{aligned}$$

we obtain the final form of (5.20) and (6.21).

Appendix B

Derivation of the Recursion for α_l

Consider $\alpha_k = \tilde{\Theta}_{m,k} \bar{R}_{m,k} \tilde{\Theta}_{m,k}^T$ and rewrite it as

$$\alpha_k = G_k \mathcal{H}_{m,k} \bar{R}_{m,k} \mathcal{H}_{m,k}^T G_k. \quad (\text{B.1})$$

Represent the product $\mathcal{H}_{m,k} \bar{R}_{m,k} \mathcal{H}_{m,k}^T$ by the sum of

$$\mathcal{H}_{m,k} \bar{R}_{m,k} \mathcal{H}_{m,k}^T = \sum_{l=0}^{N-1} (\mathcal{F}_k^{m+1+l})^{-T} H_{m+l}^T R_{m+l} H_{m+l} (\mathcal{F}_k^{m+1+l})^{-1} \quad (\text{B.2})$$

$$= H_k^T R_k H_k + F_k^{-T} \left[\sum_{l=0}^{N-2} (\mathcal{F}_{k-1}^{m+1+l})^{-T} H_{m+l}^T R_{m+l} H_{m+l} (\mathcal{F}_{k-1}^{m+1+l})^{-1} \right] F_k^{-1} \quad (\text{B.3})$$

$$= H_k^T R_k H_k + F_k^{-T} \mathcal{H}_{m,k-1} \bar{R}_{m,k-1} \mathcal{H}_{m,k-1}^T F_k^{-1}. \quad (\text{B.4})$$

Referring to $\alpha_{k-1} = G_{k-1} \mathcal{H}_{m,k-1} \bar{R}_{m,k-1} \mathcal{H}_{m,k-1}^T G_{k-1}$, find

$$\mathcal{H}_{m,k-1} \bar{R}_{m,k-1} \mathcal{H}_{m,k-1}^T = G_{k-1}^{-1} \alpha_{k-1} G_{k-1}^{-1}. \quad (\text{B.5})$$

Finally, combine (B.1), (B.4), and (B.5) and end up with the recursion (5.28) and (6.36) for α_k .

Appendix C

Derivation of the Recursion for β_k

Rewrite $\beta_k = \Theta_{m,k}^{(i)} \bar{R}_{m,k}^{(i)} \Theta_{m,k}^{(i)T}$ as

$$\beta_k = G_k^{(i)} \mathcal{H}_{m,k}^{(i)} \bar{R}_{m,k}^{(i)} \mathcal{H}_{m,k}^{(i)T} G_k^{(i)}, \quad (\text{C.1})$$

represent by the sum, and transform as

$$\mathcal{H}_{m,k}^{(i)} \bar{R}_{m,k}^{(i)} \mathcal{H}_{m,k}^{(i)T} = \sum_{l=0}^{N-1} (\mathcal{F}_k^{m+1+l})^{-T} H_{m+l}^{(i)T} R_{m+l}^{(i)} H_{m+l}^{(i)} (\mathcal{F}_k^{m+1+l})^{-1} \quad (\text{C.2})$$

$$= H_k^{(i)T} R_k^{(i)} H_k^{(i)} + F_k^{-T} \left[\sum_{l=0}^{N-2} (\mathcal{F}_{k-1}^{m+1+l})^{-T} H_{m+l}^{(i)T} R_{m+l}^{(i)} H_{m+l}^{(i)} (\mathcal{F}_{k-1}^{m+1+l})^{-1} \right] F_k^{-1} \quad (\text{C.3})$$

$$= H_k^{(i)T} R_k^{(i)} H_k^{(i)} + F_k^{-T} \mathcal{H}_{m,k-1}^{(i)} \bar{R}_{m,k-1}^{(i)} \mathcal{H}_{m,k-1}^{(i)T} F_k^{-1}. \quad (\text{C.4})$$

From $G_{k-1}^{(i)} \mathcal{H}_{m,k-1}^{(i)} \bar{R}_{m,k-1}^{(i)} \mathcal{H}_{m,k-1}^{(i)T} G_{k-1}^{(i)}$ find

$$\mathcal{H}_{m,k-1}^{(i)} \bar{R}_{m,k-1}^{(i)} \mathcal{H}_{m,k-1}^{(i)T} = G_{k-1}^{(i)-1} \beta_{k-1} G_{k-1}^{(i)-1}, \quad (\text{C.5})$$

combine (C.1), (C.4), and (C.5), and arrive at the recursion (5.29) and (6.37) for β_k .