OPTIMAL INFORMATION-WEIGHTED KALMAN CONSENSUS FILTER

A Thesis

Submitted to the Faculty

of

Purdue University

by

Shiraz Khan

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVAL

Dr. Inseok Hwang, Chair

School of Aeronautics and Astronautics

Dr. Martin J. Corless

School of Aeronautics and Astronautics

Dr. Dengfeng Sun

School of Aeronautics and Astronautics

Approved by:

Dr. Gregory A. Blaisdell

Head of the School Graduate Program

Dedicated to my parents – *Simi* & *Sheffi*.

ACKNOWLEDGMENTS

The work documented in this thesis was accomplished through the guidance and support of several people.

I would like to begin by thanking Professor Inseok Hwang, who served as my major advisor, but was above all a constant source of encouragement. Owing to his mentorship, for the first time I feel less like an outsider and more like a researcher in academia. I would also like to thank Professor Martin Corless for introducing me to linear and nonlinear system analysis, for never failing to entertain in-class questions and for being passionate about teaching the subject. I would like to thank Professor Dengfeng Sun in the same vein, who introduced me to the mathematical rigor of optimization, and has always been an approachable person.

I would like to thank Raj Deshmukh, who has not only been instrumental in the research that went into this thesis, but has been my point of contact for all things related to graduate study (and beyond). I would like to thank my friends – Rohitha, Fizaan, Govind, Navya and Simran, whose love and support kept graduate school from feeling like a solo effort, as well as Jacob, Robby, Sawyer, Joe and Tyler, for indulging me in my Saturday evening shenanigans.

Last but not the least, I am eternally grateful to my parents, who have always been there to cheer me on.

TABLE OF CONTENTS

LIST OF FIGURES

SYMBOLS

$\mathbb{R}^n$	n-dimensional real space
$\mathbb{R}^{m \times n}$	set of all $m \times n$ real-valued matrices
v^T	transpose of vector v
M^{-1}	inverse of matrix M
$E[a]$	expectation of random variable a
$E[a\|b]$	conditional expectation of variable a, given event b has occurred
δ_{ij}	Kroenecker delta, which is 1 if $i = j$ and 0 otherwise
$\\|.\\|$	2-norm operator on the Euclidean vector space
$\\|.\\|_{\mathcal{F}}$	Frobenius norm operator
$\mathcal{V} \times \mathcal{W}$	Cartesian product of sets $\mathcal{V}$ and $\mathcal{W}$
$\{x\|\Phi(x)\}$	set of all x that satisfy a logical predicate $\Phi(x)$
$\|\mathcal{V}\|$	cardinality of set $\mathcal{V}$
$col(.)$	block matrix with elements arranged as a column
$diag(.)$	block matrix with elements arranged diagonally, off diagonal elements are zero-valued
I_d	identity matrix of rank d
$A \otimes B$	Kroenecker product of A and B
$span(.)$	linear span operator
ϕ	empty set

ABBREVIATIONS

UAV	unmanned aerial vehicle
LiDAR	light detection and ranging
IoT	internet of things
KF	Kalman filter
EKF	extended Kalman filter
UKF	unscented Kalman filter
DKF	distributed Kalman filter
KCF	Kalman consensus filter
GKCF	generalized Kalman consensus filter
ICF	information weighted consensus filter
IFDKF	information driven fully distributed Kalman filter
OKCF	optimal Kalman consensus filter
OKCF-WDG	optimal Kalman consensus filter for weighted directed graphs
MMSE	minimum mean squared error
MAP	maximum a priori
FOV	field of view

ABSTRACT

Khan, Shiraz M.S., Purdue University, May 2020. Optimal Information-Weighted Kalman Consensus Filter . Major Professor: Inseok Hwang.

Distributed estimation algorithms have received considerable attention lately, owing to the advancements in computing, communication and battery technologies. They offer increased scalability, robustness and efficiency. In applications such as formation flight, where any discrepancies between sensor estimates has severe consequences, it becomes crucial to require consensus of estimates amongst all sensors. The Kalman Consensus Filter (KCF) is a seminal work in the field of distributed consensus-based estimation, which accomplishes this.

However, the KCF algorithm is mathematically sub-optimal, and does not account for the cross-correlation between the estimates of sensors. Other popular algorithms, such as the Information weighted Consensus Filter (ICF) rely on ad-hoc definitions and approximations, rendering them sub-optimal as well. Another major drawback of KCF is that it utilizes unweighted consensus, i.e., each sensor assigns equal weightage to the estimates of its neighbors. This fact has been shown to cause severely degraded performance of KCF when some sensors cannot observe the target, and can even cause the algorithm to be unstable.

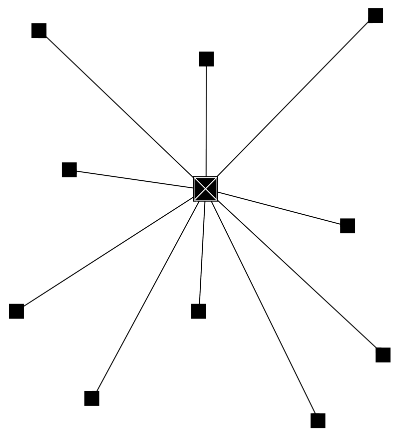
In this work, we develop a novel algorithm, which we call Optimal Kalman Consensus Filter for Weighted Directed Graphs (OKCF-WDG), which addresses both of these limitations of existing algorithms. OKCF-WDG integrates the KCF formulation with that of matrix-weighted consensus. The algorithm achieves consensus on a weighted digraph, enabling a directed flow of information within the network. This aspect of the algorithm is shown to offer significant performance improvements over KCF, as the information may be directed from well-performing sensors to other sensors which have high estimation error due to environmental factors or sensor limitations. We validate the algorithm through simulations and compare it to existing algorithms. It is shown that the proposed algorithm outperforms existing algorithms by a considerable margin, especially in the case where some sensors are naïve (i.e., cannot observe the target).

1. INTRODUCTION

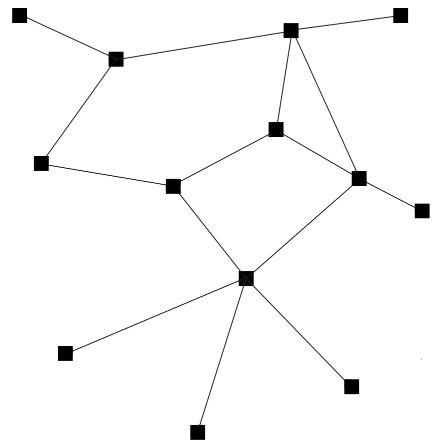
1.1 Background and Motivation

Autonomous machines have become widely popular today, due to their increased reliability and efficiency as compared to human labor. They are also immune to the fatigue that arises from repetitive tasks, as well as the errors that could result from it. These machines are often required to interface with the real world, in order to accomplish their objectives. They do so through a variety of sensors, such as cameras, accelerometers and LiDARs [1] [2]. The measurements made by these sensors invariably have noise present in them. This noise gets compounded when information is processed and transmitted through communication channels. The problem of extracting meaningful information from noisy measurements has been a popular topic in the signal processing community. Estimation algorithms achieve this by analyzing a series of measurements made over a duration of time, while incorporating the statistical properties of the noise [3].

When there are multiple sensors making observations, the redundancy from their measurements can be exploited to further filter this noise out. Such an arrangement of sensors is called a sensor network. Conventionally, the measurements of all the sensors in a network were transmitted to a central computer called a supervisor, which then does all of the processing using an estimation algorithm. Such a network architecture is said to be *centralized* (Fig. 1.1a). Centralized estimation is highly susceptible to hardware failures as well as adversarial attacks, since the performance of the network is hinged entirely on the supervisor [4] [5]. As a result of recent advancements in computation, communication and battery technology, many autonomous systems are moving towards *decentralized* architectures instead (Fig. 1.1b) - in which there are no central supervisors. Rather, in a decentralized network, the computational burden is shared by multiple (or all) agents in the network.



(a) A centralized network



(b) A decentralize network

Figure 1.1.: Two types of network architectures

The problem of distributed estimation considers a network of sensor agents where each agent can communicate with some (not necessarily all) of the other agents, make measurements, as well as carry out computational tasks. The common protocol followed by all agents in the network is called a distributed estimation algorithm. Distributed estimation algorithms have gained industrial popularity due to the arrival of new markets like autonomous vehicles and internet-of-things (IoT). It has also emerged as a popular topic in many academic fields, including computer science and controls. Distributed estimation algorithms have the added benefit of being more scalable and computationally efficient than centralized estimation algorithms.

In consensus-based distributed estimation, the sensor agents have the additional objective of arriving at the same estimate as each other. This becomes an important aspect in applications where the agents must collectively and cooperatively accomplish a global objective, such as remote sensing and target tracking [6] [7] [8], as well as in applications where discrepancies in sensor estimates can have severe consequences, such as formation flight and flocking [9] [10].

1.2 Distributed Estimation Algorithms in Literature

As centralized estimation is a relatively older topic, there is abundant literature on it [11] [12]. A popular centralized estimation algorithm is the Kalman Filter, which models the measurement noise as Gaussian processes [3]. The choice of Gaussian random variables to model uncertainties is motivated by their ease of analysis, as well as the provision of the central limit theorem, which states that the sum of independent random distributions tends to that of the Gaussian distribution. The Kalman Filter has been extended to nonlinear processes by way of the Extended Kalman Filter (EKF) [13], which linearizes the system about a point. The Unscented Kalman Filter (UKF) [14] is another popular nonlinear filter which uses a deterministic sampling technique and achieves better performance than EKF. Since centralized estimation requires all sensors to be connected to a central supervisor, it is highly vulnerable to hardware failures and cyber-attacks [4].

Distributed estimation, although a relatively newer topic, has also been well-researched. The Distributed Kalman Filter (DKF) was introduced in [15] to be used in conjunction with fully-connected sensor networks, where each sensor can communicate with all the other sensors. The DKF algorithm arrives at an accurate estimate at each sensor, but the requirement of a fully-connected network greatly constrains the scalability of the algorithm, amongst other considerations. Further work in the field relaxed this assumption, by requiring each sensor to be connected with only a subset of the other sensors [16].

Another class of distributed algorithms that has received considerable attention is that of distributed consensus [17] [18]. The objective of distributed consensus algorithms is for the estimates of all agents of a sensor network to converge to a common global value. This problem formulation can be extended to that of average consensus, where the sensors must arrive at the global average of the initial conditions. The Kalman Consensus Filter (KCF) [19] is a notable work in distributed estimation, which integrates the distributed consensus problem with a localized Kalman filter.

The resulting algorithm ensures that not only is an accurate estimate obtained at every sensor, but that the converged values of all sensors are the same. As a consequence, it is no longer required that all the sensors have full observability; The system being sensed must be collectively observable by the sensor network. Consensus based distributed estimation algorithms have been applied to target tracking [6], unmanned aerial vehicles [7] [8] and health monitoring [20], among other applications [21] [22].

The original KCF formulation required the updates across the network to be synchronous, at discrete intervals. Such synchronicity necessitates the presence of a supervisor that can coordinate all the sensors, and hence violates the conditions for an algorithm to be fully distributed, which is something that was addressed in [23]. Other successful extensions of KCF have addressed its linearity [13] [14], constraints on observability [24] [25] as well as sub-optimality [26]. The work on the Generalized KCF (GKCF) [24] considers the case of camera networks, wherein it is common for some cameras to be completely oblivious of the system being sensed, due to their limited field of view. In the presence of such sensors − called *naïve* sensors − it is demonstrated that KCF performs very poorly.

Amongst subsequent consensus-based distributed estimation algorithms [27] [28] [24], the Information weighted Consensus Filter (ICF) [25] is notable for its superior performance and robustness in presence of sensor naïvety. The authors of the Information-driven Fully Distributed Kalman Filter (IFDKF) [29] acknowledge that many existing algorithms, although claimed to be fully distributed, require global information about the sensor network, such as the maximum degree of the graph representing the sensor connectivity. The IFDKF algorithm relaxes this requirement, while still providing performance improvements in the presence of naïve sensors. Despite the popularity of GKCF, ICF and IFDKF, none of these algorithms are mathematically optimal.

1.3 Objectives and Contributions

The poor performance of KCF in the presence of naïve sensors can be attributed to the fact that a sensor utilizing KCF does not differentiate between the information received by it from neighboring sensors. As a result of this, the presence of a naïve sensor causes all sensors in its vicinity to perform poorly. As we will see in Chapter 2, the connectivity of a sensor network can be modelled as a directed graph. Under this formulation, the KCF estimation logic achieves consensus on an unweighted graph, i.e., it does not assign relative weightage to the edges. Depending on the graph topology and sensor noise characteristics, this fact can even cause the algorithm to diverge at certain (and eventually all) sensors.

The GKCF, ICF and IFDKF algorithms introduced in Section 1.2 are all notable works which address this issue, by way of weighing the information received from neighboring sensors [24] [25] [29]. By assigning a lower weightage to the information received from a naïve neighbor, a sensor may selectively choose to incorporate useful information and discard extraneous information. Furthermore, the effect of sensor naïvety is heightened for certain topologies. The path graph is an example of a network topology where weighted consensus algorithms can perform significantly better than those employing unweighted consensus.

The second major drawback of KCF is with regards to its sub-optimality, something which even subsequent algorithms such as GKCF suffer from. The KCF algorithm was designed to be in keeping with the centralized Kalman filter, which is a linear quadratic state estimator that achieves the minimum mean squared error. However, the development of KCF was facilitated by introducing approximations that enabled the author to arrive at simplified expressions [30]. Specifically, the author uses an approximation based on an order of magnitude argument. A second approximation is made in the choice of consensus gains in KCF, ultimately rendering the algorithm sub-optimal and defeating the purpose of striving towards a minimum mean squared error estimator. The authors in [26] presented the optimal KCF gains which do not rely on approximations, but the optimal KCF (OKCF) algorithm still underperforms in presence of naïve sensors, as it was developed using the framework of distributed Kalman filtering with unweighted consensus.

Similarly, the derivation of the ICF and IFDKF algorithms, although motivated by centralized Maximum A Priori (MAP) estimation, uses ad-hoc approximations which make it sub-optimal. The derivation of ICF begins with a centralized MAP estimator and extends it to the distributed case. It does so by considering the two separate cases of uncorrelated sensors and fully correlated sensors, which can be thought of as the initial and the converged states of the network respectively. The algorithm is shown to be optimal and tractable for the two aforementioned cases, but its performance during the transitional case (partially correlated sensors) is claimed to be comparable, supported only by experimental results.

Another approximation that recurs throughout the literature on distributed estimation comes from the assumption that sensors are completely uncorrelated with each other [19] [31]. The cross-covariance between the estimation errors of a pair of sensors is therefore assumed to be zero. In reality, the cross-covariances between sensors become of considerable significance when designing an optimal estimation algorithm. Since distributed estimation algorithms are iterative update rules employed at each sensor in the network, over time, some of these sensors become highly correlated. Consensus-based algorithms can only achieve optimal performance when this correlation has been taken into account, such that they may assign lower weightage to redundant information. The lack of optimality of aforementioned algorithms, as well as the poor performance of algorithms such as optimal KCF (OKCF) in the presence of sensor naïvety, motivate us to design a distributed estimation algorithm that addresses both concerns simultaneously.

The algorithm developed in this body of work accomplishes both of those objectives [32]. We present here a mathematically rigorous derivation of a distributed estimation algorithm that arrives at consensus on a weighted digraph. This is achieved by using localized Kalman filters at each sensor, and integrating them with a matrix-weighted consensus algorithm. The consensus weights thus introduced ensure that the direction of flow of information within the network is implicitly controlled.

The requirement of optimality is guaranteed as well, since this work does not rely on the ad-hoc arguments or approximations that plague existing algorithms. It instead utilizes a locally optimal formulation which results in the minimum mean squared estimation error amongst comparable distributed estimation algorithms. In order to achieve this, the error cross-covariances between the sensors are accounted for.

It is shown that the proposed algorithm outperforms other algorithms in literature, which is as expected. This is especially apparent in the presence of naïve sensors, as showcased in the simulation results.

1.4 Organization

The organization of this thesis is as follows. In Chapter 2, the problem formulation is detailed. The models describing the system and sensor network are discussed. The Kalman Consensus Filter is introduced as well, as it sets a precedent for the algorithm developed as part of this work. In Chapter 3, the proposed algorithm is derived, by solving the minimum mean squared error (MMSE) estimation problem. Chapter 4 contains further analysis of the algorithm, and derives the closed-form expressions for the equations introduced in the previous chapter. Subsequently, the information form of the algorithm is formulated, which is easier to implement and compare to existing algorithms.

In Chapter 5, the present algorithm is validated by considering a simulated scenario of a 2-dimensional dynamical system with noise, being sensed by a sensor network. A second simulation scenario considers the same dynamical system, but a special sensor network topology that helps illustrate the effects of sensor naïvety. The performance of the present algorithm is compared with those of existing algorithms. The third simulation scenario is that of a camera network, which serves as an example of a real world situation where some sensors may become naïve, and cause poor estimation performance in algorithms with unweighted consensus.

2. PROBLEM FORMULATION

In this chapter, the mathematical models of the sensors as well as the system being observed are introduced. We also present the Kalman Consensus Filter, and briefly discuss the rationale behind the algorithm.

2.1 System and Observation Models

The dynamical system being observed (called the *target*) can be represented as a discrete-time linear time-varying system,

$$x(k+1) = A(k)x(k) + B(k)w(k) \tag{2.1}$$

where $x(k) \in \mathbb{R}^n$ is the state and $w(k) \in \mathbb{R}^m$ is the system noise at time step k. $A(k)$ and $B(k)$ are the system matrices at time step k. $x(0) = x_0$ is the initial condition of the system, which is assumed to be unknown to the sensors observing it.

Assumption 1 *(Linearity) It is assumed that the system as well as the observation models of the sensors can be adequately modeled as discrete-time linear time-varying systems. In non-linear dynamical systems, this can be accomplished by linearizing the system at discrete intervals.*

The system is being observed by a network of sensors that have communication channels between them, but cannot necessarily establish all-to-all communication. The communication topology of the sensor agents (at time step k) can be modeled as a dynamic directed graph $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$. In the graph, $\mathcal{V} = \{1, 2, \ldots, N\}$ is the set of vertices, and represents the N sensor agents. We refer to the number assigned to each sensor agent as its index. The set of edges $\mathcal{E}(k) \subseteq \mathcal{V} \times \mathcal{V}$ represents the communication links.

Assumption 2 *(Two-way Communication) The communication link between any two sensors is two-way. Messages can be transmitted simultaneously in both directions along the communication link*[1].

Assumption 3 *(Instantaneous Communication) Each sensor can simultaneously and instantaneously communicate with all of its neighbors. Communication delays, although a more realistic proposition, will not be addressed in this work.*

The edge weights do not have any physical significance; Rather they model the flow of information in the sensor network, taking into consideration that some sensors might discard some of the received information on account of it being redundant.

Each sensor makes measurements in accordance with the model

$$z_i(k) = H_i(k)x(k) + v_i(k) \qquad i = 1, 2 \ldots N \tag{2.2}$$

where i denotes the index of the sensor, $z_i(k) \in \mathbb{R}^p$ is the measurement made by the sensor and $v_i(k) \in \mathbb{R}^p$ is the measurement noise.

Assumption 4 *(Collective Observability) The sensor network can collectively observe the target, i.e., the observability matrix of the entire network, constructed as,*

$$\mathcal{O} = \begin{bmatrix} H \\ HA \\ HA^2 \\ \vdots \\ HA^{n-1} \end{bmatrix}, \qquad \textit{where } H = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_N \end{bmatrix}$$

has row rank equal to n.

The system and measurement noise ($w(k)$ and $v_i(k)$, respectively) are each modeled as mutually independent white Gaussian random variables, such that

$$\begin{aligned} E[w(r)w(s)^T] &= Q(r)\delta_{rs} \\ E[v_i(r)v_j(s)^T] &= R_{i,j}(r)\delta_{rs}\delta_{ij} \end{aligned} \tag{2.3}$$

[1]Notwithstanding Assumption 2, the edge weights in either direction may be different, which admits a directional flow of information in the network. This is discussed further in Chapter 3.

where $E[\cdot]$ is the expectation operator and δ_{rs} is the Kronecker delta, i.e., $\delta_{rs} = 1$ if $r = s$, and $\delta_{rs} = 0$, otherwise.

Assumption 5 *(Uncorrelated Measurement Noise) We assume that the measurement noise of the sensors is pairwise independent. So $R_{i,j} = 0$ if $i \neq j$.*

It should be noted that Assumption 5 may often be invalid. An example of this when the measurement noise is a manifestation of environmental factors such as weather. Such a case will not be discussed here.

Assumption 6 *(State-Transition Matrix) We assume that all sensors know the state-transition matrix A, which represents the deterministic dynamics of the target, or that it can be estimated using an appropriate system identification algorithm.*

Let us denote the history of measurements made by sensor i as

$$Z_i(k) \coloneqq \{z_i(0), z_i(1), \ldots, z_i(k)\}$$

The estimates of the target at sensor i, before and after incorporating the measurement information at the current time step k, are respectively

$$\begin{aligned} \bar{x}_i(k) &= E[x(k) \mid Z_i(k-1)] \\ \hat{x}_i(k) &= E[x(k) \mid Z_i(k)] \end{aligned} \tag{2.4}$$

where $\bar{x}_i(k)$ is referred to as the prior estimate and $\hat{x}_i(k)$ as posterior estimate of the target state.

$\bar{\eta}_i = \bar{x}_i - x$ and $\hat{\eta}_i = \hat{x}_i - x$ are the prior and posterior estimation errors at sensor i, respectively. The prior and posterior cross covariance matrices of the estimation errors are

$$\begin{aligned} P_{i,j} &= E[\bar{\eta}_i \bar{\eta}_j^T] \\ M_{i,j} &= E[\hat{\eta}_i \hat{\eta}_j^T] \end{aligned} \tag{2.5}$$

respectively. The cross-covariance between two sensors ($M_{i,j}$ where $i \neq j$) is a measure of the redundancy of information between sensors. As we will see later, these quantities must be estimated as well, in order to design an optimal estimation algorithm.

2.2 Estimation Algorithm

2.2.1 Kalman Filter

The objective of an estimation algorithm is to utilize the history of measurements made over a duration to estimate the true state of the target, x. This is equivalent to requiring that the estimation error goes to zero. The Kalman filtering algorithm achieves this by making the assumption that the deterministic dynamics of the target in (2.1) is known. At every time step, it uses the measurement information $z(k)$ to update the current prior estimate $\bar{x}(k)$ of the system towards the posterior estimate $\hat{x}(k)$, as

$$\hat{x}(k) = K(k)(z(k) - H(k)\bar{x}(k)) \tag{2.6}$$

where $K(k)$ is called the Kalman gain and $H(k)$ is the observation matrix of the sensor. The evolution of the target between consecutive time steps k and $k+1$ is accounted for by propagating the prior estimate as $\bar{x}(k+1) = A\hat{x}(k)$. The optimal value of $K(k)$ which minimizes the mean squared estimation error, is

$$K = PH^T(R + HPH^T)^{-1} \tag{2.7}$$

where R is the measurement noise covariance matrix, $P = E[(\bar{x} - x)(\bar{x} - x)^T]$ and all quantities correspond to their evaluations at time step k. For brevity, we will omit the time step k from here on.

2.2.2 Kalman Consensus Filter

The Kalman Consensus Filter (KCF) [19] algorithm uses a localized Kalman filter such as (2.6) at each sensor in conjunction with a consensus term. The KCF update equation is,

$$\hat{x}_i = \bar{x}_i + K_i(z_i - H_i\bar{x}_i) + C_i \sum_{j \in \mathcal{N}_i} (\bar{x}_j - \bar{x}_i) \tag{2.8}$$

where C_i is called the consensus gain for sensor i. $\mathcal{N}_i$ is the set of indices of the neighbors of sensor i. The second term in (2.8) incorporates information from the current sensor measurement into the prior estimate. The third term drives the estimates of all sensors in a network towards consensus. In the absence of measurement noise, the state $\hat{x}_1 = \hat{x}_2 = \cdots = \hat{x}_n = x$ is a fixed point of (2.8).

The author proposes the following choice for the consensus gain C_i [30], which is chosen such that the algorithm satisfies a sufficient (but not necessary) condition for stability,

$$C_i = \epsilon \frac{P_{i,i}}{1 + \|P_{i,i}\|_F} \tag{2.9}$$

where $\| \, . \, \|_F$ is the Frobenius norm. The constant ϵ is a design parameter of the order of the discretization time step.

It is proposed by the author that the Kalman gain can be designed such that the total posterior mean squared estimation error, given by $\sum_{i=1}^{N} E[\|\hat{x}_i - x\|^2]$, is minimized. This is done by noting that the total mean squared error is equal to $\sum_{i=1}^{N} tr(M_i)$. The optimal Kalman gain for the choice of consensus gain in (2.9), is

$$K_i = P_i I_\epsilon H_i^T (R_i + H_i P_i H_i^T)^{-1} \tag{2.10}$$

where I_ϵ is a small perturbation of the identity matrix, given by

$$I_\epsilon = I + \epsilon \frac{\sum_{j \in \mathcal{N}_i} (P_{j,i} - P_{i,i})}{1 + \|P_{i,i}\|_F} \tag{2.11}$$

Furthermore, the author makes the approximation $I_\epsilon \approx I$. The resulting KCF algorithm does not account for the cross-covariances $P_{i,j}$. In [26], the authors simultaneously optimize the gains K_i and C_i of the KCF algorithm. The resulting algorithm (henceforth referred to as Optimal KCF) accounts for the cross-covariances between sensors.

3. ALGORITHM DEVELOPMENT

3.1 Extension of KCF to Weighted Consensus

The KCF and OKCF algorithms introduced in the previous chapter use unweighted consensus, which we define as follows.

Definition 3.1 ***(Unweighted Consensus)*** *Distributed estimation algorithms that can be written in the form*

$$\hat{x}_i = f_1(\bar{x}_i) + f_2(z_i) + f_3\Big(\sum_{j \in \mathcal{N}_i} (\bar{x}_j - \bar{x}_i)\Big)$$

where f_1, f_2 and f_3 are linear vector-valued functions, are said to utilize unweighted consensus.

The flow of information in a sensor network following an unweighted consensus algorithm is diffusive in nature. Unweighted consensus does not account for the case where there is significant difference between the quality of estimates of neighboring sensors. Since sensors utilizing a consensus-based algorithm communicate with each other, in the case where one of the sensors has high sensor noise, it imparts this noisy information to each of its neighbors as well.

Definition 3.2 ***(Weighted Consensus)*** *Distributed estimation algorithms are said to utilize weighted consensus, if they can be written in the form*

$$\hat{x}_i = f_1(\bar{x}_i) + f_2(z_i) + \sum_{j \in \mathcal{N}_i} g_{j,i}(\bar{x}_j - \bar{x}_i)$$

where f_1, f_2 are linear vector-valued functions, and $\{g_{j,i} | j \in \mathcal{N}_i\}$ is a set of non-identical linear functions.

Weighted consensus algorithms achieve a directed flow of information, which can be exploited to ensure that the collective performance of the network is optimal [33]. This motivates the extension of KCF towards weighted consensus.

The KCF update equation (2.8) can be generalized to incorporate weighted consensus, as

$$\hat{x}_i = \bar{x}_i + K_i(z_i - H_i\bar{x}_i) + \sum_{j\in\mathcal{N}_i}[C_{j,i}(\bar{x}_j - \bar{x}_i)] \tag{3.1}$$

where we define $C_{j,i}$ as the consensus gain for the information transmitted from sensor j to sensor i. A sensor network utilizing the update equation (3.1) can be represented as a weighted directed graph, such as the one depicted in Fig. 3.1. The consensus gain $C_{j,i}$ can be represented as the weight of the directed edge from node j to node i. The KCF algorithm (2.8) can be considered as a special case of (3.1) where $C_{m,i} = C_{n,i}$ $\forall m, n \in \mathcal{N}_i$.

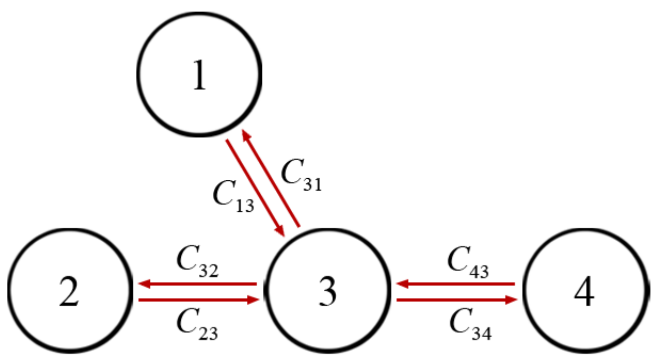


Figure 3.1.: An example of a weighted directed graph

The set of estimation gains $\{K_i, C_{j,i}|j \in \mathcal{N}_i\}$ needs to be determined. One way of choosing estimation gains is by deciding on a performance metric to be optimized. A widely accepted metric of the accuracy of a distributed estimation algorithm is the total mean squared error, defined as

$$\textit{Total Mean Squared Error} = \sum_{i=1}^{N} E[\|\hat{x}_i - x\|^2] \tag{3.2}$$

3.2 Conditions for Optimality

Theorem 3.1 *The distributed estimation algorithm following the update rule (3.1) is an optimal minimum mean squared estimator, if and only if the following conditions are satisfied,*

$$(I - K_iH_i - \sum_{s\in\mathcal{N}_i} C_{s,i})(P_{i,j} - P_{i,i}) = \sum_{s\in\mathcal{N}_i} C_{s,i}(P_{s,i} - P_{sj}) \qquad \forall j \in \mathcal{N}_i \tag{3.3}$$

$$K_i = (P_{i,i}H_i^T + \sum_{s\in\mathcal{N}_i} C_{s,i}(P_{s,i} - P_{i,i})H_i^T)\Delta_i^{-1} \tag{3.4}$$

where $\Delta_i = R_i + H_iP_{i,i}H_i^T$.

Proof To find the expression for $M_{i,j}$, let us consider the following relation between the prior and posterior estimation errors of sensors i and j. Subtracting x from both sides of (3.1) for sensors i and j,

$$\begin{aligned}\hat{\eta}_i &= F_i\bar{\eta}_i + \sum_{r\in\mathcal{N}_i}[C_{r,i}(\bar{\eta}_r - \bar{\eta}_i)] + K_iv_i \\ \hat{\eta}_j &= F_j\bar{\eta}_j + \sum_{s\in\mathcal{N}_j}[C_{s,j}(\bar{\eta}_s - \bar{\eta}_j)] + K_jv_j\end{aligned} \tag{3.5}$$

where $F_i = I - K_iH_i$. The cross covariance $M_{i,j}$ can be written as:

$$\begin{aligned}M_{i,j} = F_iP_{i,j}F_j^T &+ F_i\sum_{s\in\mathcal{N}_j} E[\bar{\eta}_i(\bar{\eta}_s^T - \bar{\eta}_j^T)C_{s,j}^T] \\ &+ \sum_{r\in\mathcal{N}_i} E[C_{r,i}(\bar{\eta}_r - \bar{\eta}_i)\bar{\eta}_j^T]F_j^T + K_iR_{i,j}K_j^T \\ &+ \sum_{r\in\mathcal{N}_i}\sum_{s\in\mathcal{N}_j} E[C_{r,i}(\bar{\eta}_r - \bar{\eta}_i)(\bar{\eta}_s^T - \bar{\eta}_j^T)C_{s,j}^T]\end{aligned} \tag{3.6}$$

For brevity, let us denote $M_{i,i}$, $P_{i,i}$ and $R_{i,i}$ as M_i, P_i and R_i, respectively. Substituting the outer product terms with the corresponding covariance matrices in (2.5), we have

$$\begin{aligned}M_i = F_iP_iF_i^T &+ \sum_{r\in\mathcal{N}_i}[C_{r,i}(P_{r,i} - P_i)]F_i^T \\ &+ F_i\sum_{s\in\mathcal{N}_i}[(P_{i,s} - P_i)C_{s,i}^T] + K_iR_iK_i^T \\ &+ \sum_{r\in\mathcal{N}_i}\sum_{s\in\mathcal{N}_i}[C_{r,i}(P_{r,s} - P_{r,i} - P_{i,s} + P_i)C_{s,i}^T]\end{aligned} \tag{3.7}$$

The total mean squared estimation error $\sum_{i=1}^{N} E[\|\hat{x}_i - x\|^2]$ can be equivalently written as $\sum_{i=1}^{N} tr(M_i)$. Considering that the gains at sensor i only influence the value of M_i, the optimal K_i and $C_{j,i}$ are the solutions to

$$\frac{\partial tr(M_i)}{\partial K_i} = 0, \qquad \frac{\partial tr(M_i)}{\partial C_{j,i}} = 0 \tag{3.8}$$

Applying the matrix trace operator $tr(\cdot)$ to (3.7), we get

$$\begin{aligned} tr(M_i) &= tr(P_i) - 2tr(P_i H_i^T K_i^T) + tr(K_i H_i P_i H_i^T K_i^T) \\ &+ 2tr[(I - K_i H_i) \sum_{s \in \mathcal{N}_i} (P_{i,s} - P_i) C_{s,i}^T] \\ &+ tr(\sum_{r \in \mathcal{N}_i} \sum_{s \in \mathcal{N}_i} [C_{r,i}(P_{r,s} - P_{r,i} - P_{i,s} + P_{i,j}) C_{s,i}^T]) \\ &+ tr(K_i R_i K_i^T) \end{aligned} \tag{3.9}$$

Taking the partial derivative of $tr(M_i)$ with respect to K_i, we can get

$$\begin{aligned} \frac{\partial tr(M_i)}{\partial K_i} &= -2P_i H_i^T + 2K_i(H_i P_i H_i^T) \\ &- 2 \sum_{r \in \mathcal{N}_i} [C_{r,i}(P_{r,i} - P_i)] H_i^T + 2K_i R_i = 0 \end{aligned} \tag{3.10}$$

From (3.10), the optimal Kalman gain K_i in terms of $C_{j,i}$ is obtained as

$$K_i = (P_i + \sum_{r \in \mathcal{N}_i} [C_{r,i}(P_{r,i} - P_i)]) H_i^T \Delta_i^{-1} \tag{3.11}$$

where $\Delta_i = R_i + H_i P_i H_i^T$. Substituting the optimal Kalman gain determined in (3.11) into (3.9), and taking the derivative with respect to $C_{j,i}$, we have

$$\begin{aligned} \frac{\partial tr(M_i)}{\partial C_{j,i}} &= (I - P_i H_i^T \Delta_i^{-1} H_i)(P_{i,j} - P_i) \\ &- \sum_{s \in \mathcal{N}_i} C_{s,i}(P_{s,i} - P_i) H_i^T \Delta_i^{-1} H_i (P_{i,j} - P_i) \\ &+ \sum_{s \in \mathcal{N}_i} C_{s,i}(P_{s,j} - P_{s,i} - P_{i,j} + P_i) = 0 \end{aligned} \tag{3.12}$$

Equation (3.12) is valid for any sensor j in $\mathcal{N}_i$. It represents a system of $|\mathcal{N}_i|$ independent linear equations in $|\mathcal{N}_i|$ unknowns, and thus can be solved uniquely.

Equation (3.11) and the system of equations (3.12) are the necessary and sufficient conditions for optimality of the present algorithm. ■

3.3 Optimal KCF for Weighted Directed Graphs

The system of equations (3.3) and (3.4) may be solved using the local information at any sensor i, as well as the set of cross-covariances $\{P_{ji}|j \in \mathcal{N}_i\}$. One way of accomplishing this is to rewrite (3.4) in matrix form, and carry out a matrix inversion to obtain the set of optimal consensus gains $\{C_{ji}|j \in \mathcal{N}_i\}$.

To write these matrices out mathematically, we wish to represent the estimation gains of sensor i in a block matrix form. Let us assign an arbitrary ordering to the set of neighbors $\mathcal{N}_i \subset \mathcal{V}$ [1], such that it is an ordered list of elements, also called a tuple. We define the function $\mathcal{N}_i(k) : 1, 2 \ldots |\mathcal{N}_i| \to \mathcal{N}_i$, which maps the positive integer k to the k^{th} element in the tuple of vertices $\mathcal{N}_i$.

Now let $\mathcal{B} = col(\mathcal{B}_1, \mathcal{B}_2 \ldots \mathcal{B}_{|\mathcal{N}_i|})$ be a block column matrix, and $\mathcal{A}$ be a block matrix, of the form

$$\begin{aligned} \mathcal{B}_r &= (I - P_{i,i} H_i^T \Delta_i^{-1} H_i)(P_{i,\mathcal{N}_i(r)} - P_{i,i}) \\ \mathcal{A}_{r,s} &= (P_{\mathcal{N}_i(r),i} - P_{i,i}) H_i^T \Delta_i^{-1} H_i (P_{i,\mathcal{N}_i(s)} - P_{i,i}) \\ &\quad - [P_{\mathcal{N}_i(r),\mathcal{N}_i(s)} - P_{\mathcal{N}_i(r),i} - P_{i,\mathcal{N}_i(s)} + P_{i,i}] \end{aligned} \tag{3.13}$$

then the consensus gains can be obtained as,

$$\begin{bmatrix} C_{\mathcal{N}_i(1),i} & C_{\mathcal{N}_i(2),i} & \ldots & C_{\mathcal{N}_i(|\mathcal{N}_i|),i} \end{bmatrix} = \mathcal{B}\mathcal{A}^{-1} \tag{3.14}$$

Finally, the Kalman gain K_i may be computed as a function of the consensus gains.

The resulting optimal distributed state estimation algorithm is summarized in Algorithm 1, which we call the *Optimal Kalman Consensus Filter for Weighted Directed Graphs* (KCF-WDG).

---

[1] The rest of the analysis is independent of this ordering, so there is no loss of generality.

Algorithm 1 Optimal KCF for Weighted Directed Graphs

Given : $\mathcal{N}_i$; $P_{i,j}(0) = P_{i,j,0}\ \forall\ i, j$; $\bar{x}_i(0) = \bar{x}_{i,0}\ \forall\ i$; $x(0) = x_0$;

At sensor i, time step k,

1: The measurement z_i is made.

2: Prior estimate $\bar{x}_i$ and cross covariance matrices $P_{i,j}$ are sent to neighboring sensors $j \in \mathcal{N}_i$.

3: Information received from the neighboring sensors is assimilated to obtain the consensus gains, by constructing the block matrices in (3.13) and carrying out the matrix inversion presented in (3.14). The Kalman gains K_i are computed as a function of consensus gains, as

$$\begin{aligned} K_i &= (P_i + \sum_{r\in\mathcal{N}_i} [C_{r,i}(P_{r,i} - P_i)])H_i^T \Delta_i^{-1} \\ \Delta_i &= R_i + H_i P_i H_i^T \end{aligned}$$

4: The computed gains are used to determine the posterior estimate ($\hat{x}_i$) of sensor i, given by

$$\hat{x}_i = \bar{x}_i + K_i(z_i - H_i\bar{x}_i) + \sum_{j\in\mathcal{N}_i} C_{j,i}(\bar{x}_j - \bar{x}_i)$$

The posterior covariance matrices are computed using

$$\begin{aligned} M_{i,j} = F_i P_{i,j} F_j^T &+ \sum_{r\in\mathcal{N}_i} C_{r,i}(P_{r,j} - P_{i,j})F_j^T \\ &+ F_i \sum_{s\in\mathcal{N}_j} (P_{i,s} - P_{i,j})C_{s,j}^T + K_i R_{i,j} K_j^T \\ + \sum_{r\in\mathcal{N}_i}\sum_{s\in\mathcal{N}_i} &[C_{r,i}(P_{r,s} - P_{r,j} - P_{i,s} + P_{i,j})C_{s,j}^T] \end{aligned}$$

5: The values of $P_{i,j}$ and $\bar{x}_i$ for the next timestep are obtained by propagating the posterior quantities, as

$$\begin{aligned} P_{i,j} &\leftarrow A M_{i,j} A^T + BQB^T \\ \bar{x}_i &\leftarrow A\hat{x}_i \end{aligned}$$

4. FURTHER ANALYSIS AND OPTIMALITY

In the last chapter, we derived the OKCF-WDG algorithm, as well as the necessary and sufficient conditions for its optimality. In this chapter, we will obtain the closed-form expressions for the estimation gains which satisfy these conditions. To be able to do so, we first rewrite the estimation gains in a vector form.

4.1 Estimation Gain Vector

The estimate propagation equation (3.1) can be equivalently written as

$$\hat{x}_i = K_i z_i + (I - K_i H_i - \sum_{j \in \mathcal{N}_i} C_{j,i})\bar{x}_i + \sum_{j \in \mathcal{N}_i} C_{j,i}\bar{x}_j \tag{4.1}$$

The coefficient in the second term of (4.1) can be interpreted as the gain for the prior estimate of sensor i. This motivates us to define

$$C_{i,i} = I - K_i H_i - \sum_{j \in \mathcal{N}_i} C_{j,i} \tag{4.2}$$

The set of gains[1] to be determined at sensor i can be written as the block-matrix

$$\mathcal{C}_i = \begin{bmatrix} C_{\mathcal{N}_i(1),i} & C_{\mathcal{N}_i(2),i} & \dots & C_{\mathcal{N}_i(|\mathcal{N}_i|),i} & C_{i,i} \end{bmatrix} \tag{4.3}$$

where $\mathcal{C}_i \in \mathbb{R}^{n \times n(|\mathcal{N}_i|+1)}$. Since $\mathcal{C}_i$ resembles a row-vector, we will henceforth refer to it as the *estimation gain vector*.

Accordingly, define the vector of estimation errors as

$$\bar{\mathbb{n}}_i = \begin{bmatrix} \bar{\eta}^T_{\mathcal{N}_i(1)} & \bar{\eta}^T_{\mathcal{N}_i(2)} & \dots & \bar{\eta}^T_{\mathcal{N}_i(|\mathcal{N}_i|)} & \bar{\eta}^T_i \end{bmatrix}^T \tag{4.4}$$

[1] Note that we still have $|\mathcal{N}_i| + 1$ independent gains in total, as C_{ii} can be uniquely determined as a function of the other gains.

It is demonstrated in Theorem 4.1 that the covariance of this vector is of significance, and is defined as

$$\mathcal{P}_i = E[\bar{\mathbb{n}}_i \bar{\mathbb{n}}_i^T] \tag{4.5}$$

which can be equivalently computed as

$$\mathcal{P}_i = \begin{bmatrix} P_{\mathcal{N}_i(1),\mathcal{N}_i(1)} & P_{\mathcal{N}_i(1),\mathcal{N}_i(2)} & \dots & P_{\mathcal{N}_i(1),\mathcal{N}_i(|\mathcal{N}_i|)} & P_{\mathcal{N}_i(1),i} \\ P_{\mathcal{N}_i(2),\mathcal{N}_i(1)} & & & & P_{\mathcal{N}_i(2),i} \\ \vdots & & \ddots & & \vdots \\ P_{\mathcal{N}_i(|\mathcal{N}_i|),\mathcal{N}_i(1)} & & & & P_{\mathcal{N}_i(|\mathcal{N}_i|),i} \\ P_{i,\mathcal{N}_i(1)} & P_{i,\mathcal{N}_i(2)} & \dots & P_{i,\mathcal{N}_i(|\mathcal{N}_i|)} & P_{i,i} \end{bmatrix} \tag{4.6}$$

Theorem 4.1 *(Necessary Condition for Optimality)*

If the distributed estimation protocol following the update rule (3.1) minimizes the total mean-squared error defined in (3.2), then the estimation gain vectors have the form

$$\mathcal{C}_i = \tilde{C}_i \mathbb{1}^T \mathcal{P}_i^{-1} \qquad \forall i \in \mathcal{V} \tag{4.7}$$

for some $\tilde{C}_i \in \mathbb{R}^{n \times n}$, where

$$\mathbb{1} = \left(\begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} \otimes I_n \right)^T$$

I_n is the identity matrix of rank n and '$\otimes$' is the Kroenecker product operator.

Proof Substituting the definition for $C_{i,i}$ (4.2) in the first optimality condition (3.3) at sensor i, we get

$$\sum_{s \in \mathcal{N}_i'} C_{si}(P_{si} - P_{sj}) = 0 I_n \qquad j \in \mathcal{N}_i \tag{4.8}$$

where $\mathcal{N}_i' = \mathcal{N}_i \bigcup \{i\}$. The system of equations (4.8) can be written in a matrix form as

$$\mathcal{C}_i \begin{bmatrix} P_{\mathcal{N}_i(1),i} - P_{\mathcal{N}_i(1),\mathcal{N}_i(1)} & \dots & P_{\mathcal{N}_i(1),i} - P_{\mathcal{N}_i(1),\mathcal{N}_i(|\mathcal{N}_i|)} & P_{\mathcal{N}_i(1),i} - P_{\mathcal{N}_i(1),i} \\ \vdots & & & \vdots \\ P_{\mathcal{N}_i(|\mathcal{N}_i|),i} - P_{\mathcal{N}_i(|\mathcal{N}_i|),\mathcal{N}_i(1)} & & & P_{\mathcal{N}_i(|\mathcal{N}_i|),i} - P_{\mathcal{N}_i(|\mathcal{N}_i|),i} \\ P_{i,i} - P_{i,\mathcal{N}_i(1)} & \dots & P_{i,i} - P_{i,\mathcal{N}_i(|\mathcal{N}_i|)} & P_{i,i} - P_{i,i} \end{bmatrix} = \mathbb{0}^T \tag{4.9}$$

where

$$\mathbb{0} = \left(\begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix} \otimes I_n \right)^T$$

Note that the matrix in (4.9) is rank-deficient, since the last n columns are zero-valued. These columns represent the equation obtained by setting $j = i$ in (4.8), which is trivially satisfied by any choice of gains. Since, for any matrix, column-rank equals row-rank, the matrix in (4.9) has a non-trivial left null-space, which in turn guarantees the existence of a non-zero estimation gain vector $\mathcal{C}_i$ satisfying the optimality condition.

Equation (4.8) can be written as

$$\mathcal{C}_i \left(\begin{bmatrix} P_{\mathcal{N}_i(1),i} \\ P_{\mathcal{N}_i(2),i} \\ \vdots \\ P_{\mathcal{N}_i(|\mathcal{N}_i|),i} \\ P_{i,i} \end{bmatrix} \mathbb{1}^T - \mathcal{P}_i \right) = \mathbb{0}^T \tag{4.10}$$

From the definition of $\mathcal{P}_i$, we have

$$\begin{bmatrix} P_{\mathcal{N}_i(1),i} \\ P_{\mathcal{N}_i(2),i} \\ \vdots \\ P_{\mathcal{N}_i(|\mathcal{N}_i|),i} \\ P_{i,i} \end{bmatrix} = \mathcal{P}_i \left(\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \otimes I_n \right) \tag{4.11}$$

Substituting (4.11) in (4.10), we get

$$\mathcal{C}_i(\mathcal{P}_i \left(\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \otimes I_n \right) \mathbb{1}^T - \mathcal{P}_i) = \mathbb{0}^T \tag{4.12}$$

The Kroenecker product is associative, and satisfies, for matrices A_1, A_2, B_1 and B_2 of appropriate dimensions,

$$(A_1 \otimes A_2)(B_1 \otimes B_2) = (A_1 B_1) \otimes (A_2 B_2)$$

which is known as the mixed-product property. Using this property in (4.12), we get

$$\mathcal{C}_i \mathcal{P}_i \left(\begin{bmatrix} -1 & 0 & \dots & 0 & 0 \\ 0 & -1 & \dots & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & \dots & -1 & 0 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix} \otimes I_n \right) = \mathbb{0}^T \tag{4.13}$$

The left null-space of the matrix in (4.13) is one-dimensional. Therefore, we have the equality

$$ker_L \left(\begin{bmatrix} -1 & 0 & \dots & 0 & 0 \\ 0 & -1 & \dots & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & \dots & -1 & 0 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix} \right) = span(\begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}) \tag{4.14}$$

where $ker_L(A) = \{x | x^T A = 0, x \in \mathbb{R}^m\}$. So the optimality condition (4.13) is equivalent to

$$\mathcal{C}_i \mathcal{P}_i = \tilde{C}_i(\begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} \otimes I_n) \tag{4.15}$$

Since $\mathcal{P}_i$ is a symmetric positive-definite matrix, it is invertible. This completes the proof. ■

Going back to the analogous comparison of $\mathcal{C}_i$ to a vector, Theorem 4.1 gives us the optimal direction for this vector. The optimal matrix-valued magnitude for this vector may be obtained utilizing the second condition (3.4) in Theorem 3.1.

Theorem 4.2 *(Necessary and Sufficient Condition for Optimality)*
The distributed estimation protocol following the update rule (3.1) minimizes the total mean-squared error defined in (3.2) if and only if

$$\mathcal{C}_i = \tilde{C}_i \mathbb{1}^T \mathcal{P}_i^{-1} \qquad \forall i \in \mathcal{V} \tag{4.16}$$

where

$$\tilde{C}_i = (\mathbb{1}^T \mathcal{P}_i^{-1} \mathbb{1} + H_i^T R_i^{-1} H_i)^{-1} \tag{4.17}$$

Proof Substituting (4.2) in the second optimality condition (3.4), we have

$$\sum_{s \in \mathcal{N}_i'} C_{s,i}[I + (P_{s,i} - P_{i,i}) H_i^T \Delta_i^{-1} H_i] = I - P_{ii} H_i^T \Delta_i^{-1} H_i \tag{4.18}$$

where $\Delta_i = R_i + H_i P_{i,i} H_i^T$. Rewriting (4.18) in terms of the estimation gain vector,

$$\mathcal{C}_i \left(\begin{bmatrix} I - P_{i,i} H_i^T \Delta_i^{-1} H_i \\ I - P_{i,i} H_i^T \Delta_i^{-1} H_i \\ \vdots \\ I - P_{i,i} H_i^T \Delta_i^{-1} H_i \end{bmatrix} + \begin{bmatrix} P_{\mathcal{N}_i(1),i} H_i^T \Delta_i^{-1} H_i \\ P_{\mathcal{N}_i(2),i} H_i^T \Delta_i^{-1} H_i \\ \vdots \\ P_{\mathcal{N}_i(|\mathcal{N}_i|),i} H_i^T \Delta_i^{-1} H_i \end{bmatrix} \right) = I - P_{i,i} H_i^T \Delta_i^{-1} H_i \tag{4.19}$$

Substituting (4.11) in (4.19), we get

$$\mathcal{C}_i \left(\mathbb{1}(I - P_{i,i} H_i^T \Delta_i^{-1} H_i) + \mathcal{P}_i \left(\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \otimes I_n \right) H_i^T \Delta_i^{-1} H_i \right) = I - P_{i,i} H_i^T \Delta_i^{-1} H_i \tag{4.20}$$

We now utilize the result of Theorem 4.1. Substituting the estimation gain vector with its optimal expression (4.7),

$$\tilde{C}_i \left(\mathbb{1}^T \mathcal{P}_i^{-1} \mathbb{1} (I - P_{i,i} H_i^T \Delta_i^{-1} H_i) + \mathbb{1}^T \left(\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \otimes I_n \right) H_i^T \Delta_i^{-1} H_i \right) = I - P_{i,i} H_i^T \Delta_i^{-1} H_i \tag{4.21}$$

By invoking the mixed-product property of Kroenecker products once again, we get

$$\tilde{C}_i \left(\mathbb{1}^T \mathcal{P}_i^{-1} \mathbb{1} (I - P_{i,i} H_i^T \Delta_i^{-1} H_i) + H_i^T \Delta_i^{-1} H_i \right) = I - P_{i,i} H_i^T \Delta_i^{-1} H_i \tag{4.22}$$

which simplifies to give,

$$\tilde{C}_i = (\mathbb{1}^T \mathcal{P}_i^{-1} \mathbb{1} + ((H_i^T \Delta_i^{-1} H_i)^{-1} - P_{i,i})^{-1})^{-1} \tag{4.23}$$

We can simplify this further by recalling the Woodbury Matrix Identity,

$$(A + UCV)^{-1} = A^{-1} - A^{-1} U (C^{-1} - V A^{-1} U)^{-1} V A^{-1}$$

Using this identity, equation (4.23) can be reduced to (4.17), thus completing our proof. ■

4.2 Information Form of OKCF-WDG

Consider the matrix $\mathcal{P}$ constructed as

$$\mathcal{P} = \begin{bmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,N} \\ P_{2,1} & P_{2,2} & & \vdots \\ \vdots & & \ddots & \\ P_{N,1} & \dots & & P_{N,N} \end{bmatrix} \tag{4.24}$$

The inverted matrix $\mathcal{F} = \mathcal{P}^{-1}$ is referred to as the *information matrix*, which shows up quite often in the literature on distributed estimation. The information matrix is (in a vague sense) proportional to the collective information possessed by the network.

In our proposed algorithm, the quantity $\mathcal{F}_i = \mathcal{P}_i^{-1}$ is of interest, as noted in Theorems 4.1 and 4.2. We will refer to this quantity as the *distributed information matrix*.

Let $\mathcal{F}_i = [F_{r,s}]$ be a block matrix (for convenience, we let r and s take the same values as the indices in $\mathcal{P}_i$, though it must be noted that $F_{r,s}$ is not solely a function of $P_{r,s}$). We then have the expressions,

$$\mathbb{1}^T \mathcal{P}_i^{-1} = \mathbb{1}^T \mathcal{F}_i = \left[\sum_{r \in \mathcal{N}_i'} F_{r,\mathcal{N}_i(1)} \quad \sum_{r \in \mathcal{N}_i'} F_{r,\mathcal{N}_i(2)} \quad \cdots \quad \sum_{r \in \mathcal{N}_i'} F_{r,|\mathcal{N}_i|} \quad \sum_{r \in \mathcal{N}_i'} F_{r,i} \right] \tag{4.25}$$

and

$$\mathbb{1}^T \mathcal{P}_i^{-1} \mathbb{1} = \mathbb{1}^T \mathcal{F}_i \mathbb{1} = \sum_{r \in \mathcal{N}_i'} \sum_{s \in \mathcal{N}_i'} F_{r,s} \tag{4.26}$$

Substituting these in our expression for the estimation gain vector, the optimal consensus gain at sensor i, corresponding to a neighboring sensor $j \in \mathcal{N}_i$, is

$$C_{j,i} = (\sum_{r \in \mathcal{N}_i'} \sum_{s \in \mathcal{N}_i'} F_{r,s} + H_i^T R_i^{-1} H_i)^{-1} \sum_{r \in \mathcal{N}_i'} F_{r,j} \tag{4.27}$$

and the Kalman gain can thereafter be determined using

$$K_i = (\sum_{r \in \mathcal{N}_i'} \sum_{s \in \mathcal{N}_i'} F_{r,s} + H_i^T R_i^{-1} H_i)^{-1} H_i^T R_i^{-1} \tag{4.28}$$

Also, from (4.1), we have the following expression for the estimation error,

$$\hat{\eta}_i = \mathcal{C}_i \bar{\mathbb{n}}_i \tag{4.29}$$

which gives the following expression for the cross-covariances,

$$M_{i,j} = \tilde{C}_i \mathbb{1}^T \mathcal{P}_i^{-1} \begin{bmatrix} P_{\mathcal{N}_i(1),\mathcal{N}_j(1)} & P_{\mathcal{N}_i(1),\mathcal{N}_j(2)} & \cdots & P_{\mathcal{N}_i(1),\mathcal{N}_j(|\mathcal{N}_j|)} & P_{\mathcal{N}_i(1),j} \\ P_{\mathcal{N}_i(2),\mathcal{N}_j(1)} & & & & P_{\mathcal{N}_i(2),j} \\ \vdots & & \ddots & & \vdots \\ P_{\mathcal{N}_i(|\mathcal{N}_i|),\mathcal{N}_j(1)} & & & & P_{\mathcal{N}_i(|\mathcal{N}_i|),j} \\ P_{i,\mathcal{N}_j(1)} & P_{i,\mathcal{N}_j(2)} & \cdots & P_{i,\mathcal{N}_j(|\mathcal{N}_j|)} & P_{i,j} \end{bmatrix} \mathcal{P}_j^{-1} \mathbb{1} \tilde{C}_j \tag{4.30}$$

or

$$M_{i,j} = \tilde{C}_i \sum_{r \in \mathcal{N}_i'} \sum_{t \in \mathcal{N}_j'} \Big((\sum_{s \in \mathcal{N}_i'} F_{r,s}) P_{r,t} (\sum_{s \in \mathcal{N}_j'} F_{t,s}) \Big) \tilde{C}_j \tag{4.31}$$

Thus we arrive at the information form of OKCF-WDG, as summarized in Algorithm 2.

Algorithm 2 Information Form of Optimal KCF-WDG

Given : $\mathcal{N}_i$; $P_{i,j}(0) = P_{i,j,0}\ \forall\ i, j$; $\bar{x}_i(0) = \bar{x}_{i,0}\ \forall\ i$; $x(0) = x_0$;

At sensor i, time step k,

1: The measurement z_i is made.

2: Prior estimate $\bar{x}_i$ and cross covariance matrices $P_{i,j}$ are sent to neighboring sensors $j \in \mathcal{N}_i$.

3: The information matrix $\mathcal{F}_i = \mathcal{P}_i^{-1}$ is computed at sensor i.

4: The estimation gains at sensor i are

$$C_{j,i} = (\sum_{r\in\mathcal{N}_i'}\sum_{s\in\mathcal{N}_i'} F_{r,s} + H_i^T R_i^{-1} H_i)^{-1} \sum_{r\in\mathcal{N}_i'} F_{r,j}$$

$$K_i = (\sum_{r\in\mathcal{N}_i'}\sum_{s\in\mathcal{N}_i'} F_{r,s} + H_i^T R_i^{-1} H_i)^{-1} H_i^T R_i^{-1}$$

5: The computed gains are used to determine the posterior estimate $(\hat{x}_i)$ of sensor i, given by

$$\hat{x}_i = \bar{x}_i + K_i(z_i - H_i\bar{x}_i) + \sum_{j\in\mathcal{N}_i} C_{j,i}(\bar{x}_j - \bar{x}_i)$$

The posterior covariance matrices are computed using

$$M_{i,j} = \tilde{C}_i \sum_{r\in\mathcal{N}_i'}\sum_{t\in\mathcal{N}_j'} \Big((\sum_{s\in\mathcal{N}_i'} F_{r,s}) P_{r,t} (\sum_{s\in\mathcal{N}_j'} F_{t,s}) \Big) \tilde{C}_j$$

$$\tilde{C}_i = (\sum_{r\in\mathcal{N}_i'}\sum_{s\in\mathcal{N}_i'} F_{r,s} + H_i^T R_i^{-1} H_i)^{-1}$$

$$\tilde{C}_j = (\sum_{r\in\mathcal{N}_j'}\sum_{s\in\mathcal{N}_j'} F_{r,s} + H_j^T R_j^{-1} H_j)^{-1}$$

6: The values of $P_{i,j}$ and $\bar{x}_i$ for the next timestep are obtained by propagating the posterior quantities, as

$$P_{i,j} \leftarrow A M_{i,j} A^T + BQB^T$$

$$\bar{x}_i \leftarrow A\hat{x}_i$$

4.3 Comparison with Existing Algorithms

In this section, we draw parallels between the Optimal KCF-WDG algorithm and other distributed estimation algorithms in literature.

Maximum a priori (MAP) estimation is a Bayesian approach to the estimation problem. The centralized MAP estimate $\hat{x}_{\text{MAP}}$, for the case where all sensors are connected to a central computer, is given by

$$\hat{x}_{\text{MAP}} = \Big(\sum_{r\in\mathcal{V}} \sum_{s\in V} F_{r,s} + \sum_{r\in\mathcal{V}} H_r^T R_r^{-1} H_r \Big)^{-1} \Big(\sum_{i\in\mathcal{V}} \sum_{r\in\mathcal{V}} F_{r,i}\bar{x}_i + \sum_{i\in\mathcal{V}} H_i^T R_i^{-1} z_i \Big) \quad (4.32)$$

where we have followed the notation used in this work. Here, we call $\mathcal{F} = [F_{r,s}] = \mathcal{P}^{-1}$ the centralized prior information matrix.

The Information weighted Consensus Filter (ICF) is derived by starting with the centralized MAP and extending it to the distributed case, by approximating the centralized prior information matrix $\mathcal{P}^{-1}$. The ICF algorithm uses sub-iterations to arrive at consensus on the normalization factor in (4.32), which is $\Big(\sum_{r\in\mathcal{V}} \sum_{s\in V} F_{r,s} + \sum_{r\in\mathcal{V}} H_r^T R_r^{-1} H_r \Big)$.

With these characteristics of existing algorithms in mind, we can draw comparisons of these algorithms to OKCF-WDG,

- The normalization factor in MAP (4.32) is the centralized equivalent of the normalization factor in OKCF-WDG (4.27). The gains in MAP are also computed similarly to OKCF-WDG, through an inversion of the centralized information matrix.

- The ICF algorithm assumes that the centralized prior information matrix must be estimated in order to achieve best estimation performance. However, we have shown in this work that it is sufficient (and possibly necessary) to utilize the distributed information matrix $\mathcal{P}_i^{-1}$ instead, in order to achieve the optimal estimation performance.

- The MAP, ICF and OKCF-WDG algorithms update the posterior estimate towards a convex combination of the information available locally. In other words, the sum of weights assigned to the prior estimate, current measurement and neighbors' estimates equal to I_n. The same is not true for KCF. This is apparent when one considers the information form of KCF; The normalization factor in KCF is $(P_{i,i}^{-1} + H_i^T R_i^{-1} H_i)$, whereas the consensus gain depends on an arbitrary design parameter ϵ.

5. VALIDATION AND BENCHMARKING OF ALGORITHM

In this chapter, we present some simulation results which validate the algorithm developed in this work, and showcase the improvements that it offers over existing algorithms.

5.1 Scenario I : Fully Connected Network

As a preliminary demonstration of the proposed algorithm, we consider the case where a fully connected network of 6 sensors (shown in Fig. 5.1) observes a 2-dimensional target.

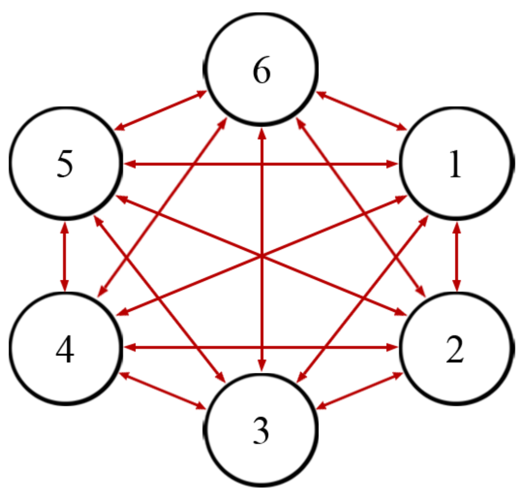


Figure 5.1.: Scenario I – Fully connected network

The target traverses a circular path, in presence of system noise. The resulting trajectory is approximately circular with perturbations. The motion of the target can be represented by the dynamical system in (2.1), with the system matrices

$$A = \begin{bmatrix} \cos(\pi/200) & -\sin(\pi/200) \\ \sin(\pi/200) & \cos(\pi/200) \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.1}$$

The initial condition is $x_0 = \begin{bmatrix} 20 & 0 \end{bmatrix}^T$.

In this scenario, each sensor can fully observe the target. The observation model of the sensors is given by (2.2). The sensor noise covariance and system noise covariance are

$$R_{i,j} = \begin{bmatrix} \delta_{ij} & 0 \\ 0 & \delta_{ij} \end{bmatrix} \qquad Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.2}$$

respectively.

The target is simulated for 500 time steps, while the sensor network utilizes the OKCF-WDG algorithm to estimate the target. In Fig. 5.2, the trajectory of the target is plotted, overlaid with the estimate at sensor 3.

The Kalman and consensus gains (K_i and $C_{j,i}$ respectively) are functions of the estimated error covariances, and are therefore time-varying quantities. However, since the deterministic dynamics of the target, system noise and observation matrices are time-invariant, the estimation gains approach a steady-state value after sufficient time steps. The values of the estimation gains of OKCF-WDG, at time step 500, are

$$K_i = 0.565 I_2 \quad C_{j,i} = 0.0725 I_2 \quad \forall i, j \in \{1, 2 \ldots 6\}$$

where I_2 is the identity matrix of order 2.

For comparison, the simulation was also run with the OKCF algorithm for 500 time steps. The steady-state values of the gains in the OKCF algorithm were found to be

$$K_i = 0.565 I_2 \quad C_i = 0.0725 I_2 \quad \forall i \in \{1, 2 \ldots 6\}$$

which are identical to the corresponding gain values in the OKCF-WDG simulation. This is because all the sensors in the network in Fig. 5.1 have statistically identical

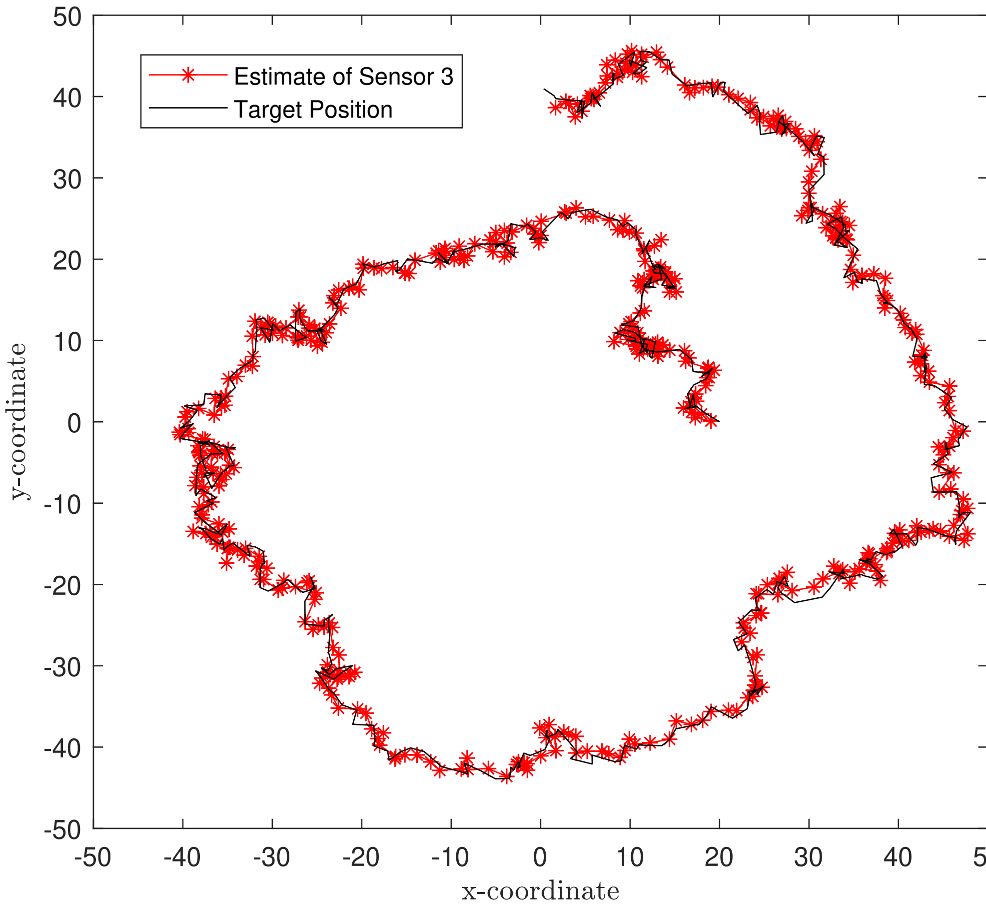


Figure 5.2.: **Scenario I** – Trajectory of the target overlaid with the estimate of sensor 3, for the fully connected sensor network shown in Fig. 5.1, using the OKCF-WDG algorithm.

observation models, and each node of the graph has the same degree. The steady-state performance of the proposed algorithm is therefore consistent with that of the OKCF scheme. That is to say, the steady-state values of consensus gains in OKCF-WDG equal that of the single consensus gain in OKCF, when all sensors are statistically identical.

5.2 Scenario II : Naïve Sensors

The previous simulation scenario used a fully connected network, and was shown to be a degenerate case that exhibits identical estimation performance with either of unweighted and weighted consensus algorithms. In this simulation we consider the chain topology (shown in Fig. 5.3), which is a path graph of 6 sensors connected serially. The target follows the same dynamical model as in scenario 1.

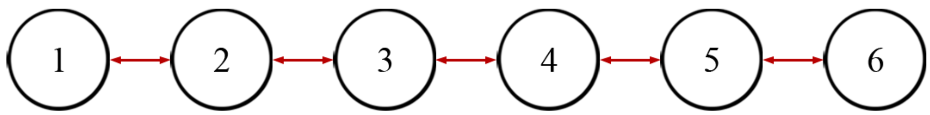


Figure 5.3.: Scenario II – Chain topology

Letting $\mathcal{N}_{Naïve}(k)$ denote the set of naïve sensors at time step k, we define that $i \in \mathcal{N}_{Naïve}(k)$ if and only if sensor i cannot observe the target altogether at time step k, i.e., its measurements contain no useful information. The simulation duration is 60 time steps, during which the set of naïve sensors switches as

$$\mathcal{N}_{Naïve}(k) = \begin{cases} \phi, & k \in [0, 20) \bigcup [40, 60] \\ \{4, 5, 6\}, & k \in [20, 40) \end{cases}$$

where ϕ is the empty set. So, from time steps 40 to 60, the target becomes oblivious to the right half of the sensor network in Fig. 5.3. Such a scenario can arise when some sensors in the network malfunction, due to hardware failures or cyber attacks. It can also occur due to limited sensing range.

For the purpose of the simulation, the effect of sensor naïvety is imposed by varying the noise covariances of corresponding sensors,

$$R_i(k) = \begin{cases} I_2, & i \notin \mathcal{N}_{Na\ddot{i}ve}(k) \\ 10^6 I_2, & i \in \mathcal{N}_{Na\ddot{i}ve}(k) \end{cases}$$

In Fig. 5.4, the mean squared estimation error is plotted for the sub-optimal KCF, OKCF and OKCF-WDG algorithms, averaged over 10,000 statistically identical Monte Carlo simulations. Between time steps $20-40$ (shaded region in Fig. 5.4, where sensors 4, 5 and 6 are naïve) the mean squared estimation error of KCF is seen to increase drastically. The OKCF-WDG algorithm achieves the lowest mean squared error in this region, and remains saturated at this value. When the system becomes observable at all sensors, the performance of all algorithms becomes comparable once again.

Figure 5.5 shows the consensus gains computed at sensor 4 over the duration of the simulation. Since these gains are matrix valued, we compare their Frobenius norms. In the OKCF-WDG algorithm, the consensus gains $C_{3,4}$ and $C_{5,4}$ are almost equal during time steps $0 - 20$ and $40 - 60$, when all sensors can observe the target. This is akin to what was observed in the previous simulation scenario. During time steps $20 - 40$, sensor 4 becomes naïve. It has one neighbor that is also naïve (sensor 5) and one neighbor that can fully observe the target (sensor 3). Consequently, the OKCF-WDG algorithm assigns higher weightage to information received from sensor 3 ($C_{3,4}$) and reduces the weightage given to sensor 5 ($C_{5,4}$).

On the other hand, Optimal KCF has a single consensus gain C_4 at sensor 4, which is used to weigh the information coming from sensors 3 and 5. During time steps $20 - 40$, the value of C_4 increases, and is seen to exhibit a chattering behavior thereafter. As a result, at sensor 4, the amount of useful information incorporated from sensor 3 is greater during this duration, as is the amount of noisy information incorporated from sensor 5. The combination of these two conflicting effects is what is observed in the estimation error dynamics presented in Fig. 5.4 $-$ leading to a net degradation of estimation performance.

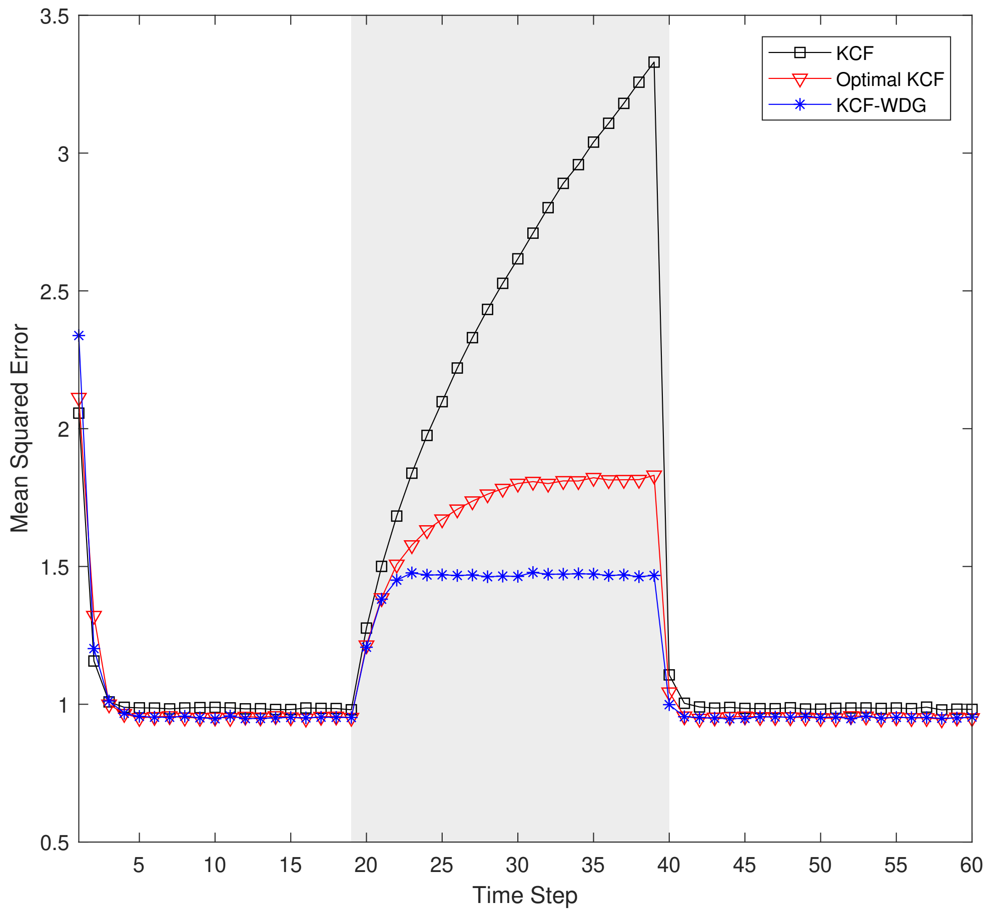


Figure 5.4.: **Scenario II** – Mean squared estimation error (averaged over 10,000 simulations) where the shaded region indicates the time steps where sensors 4, 5 and 6 cannot observe the target.

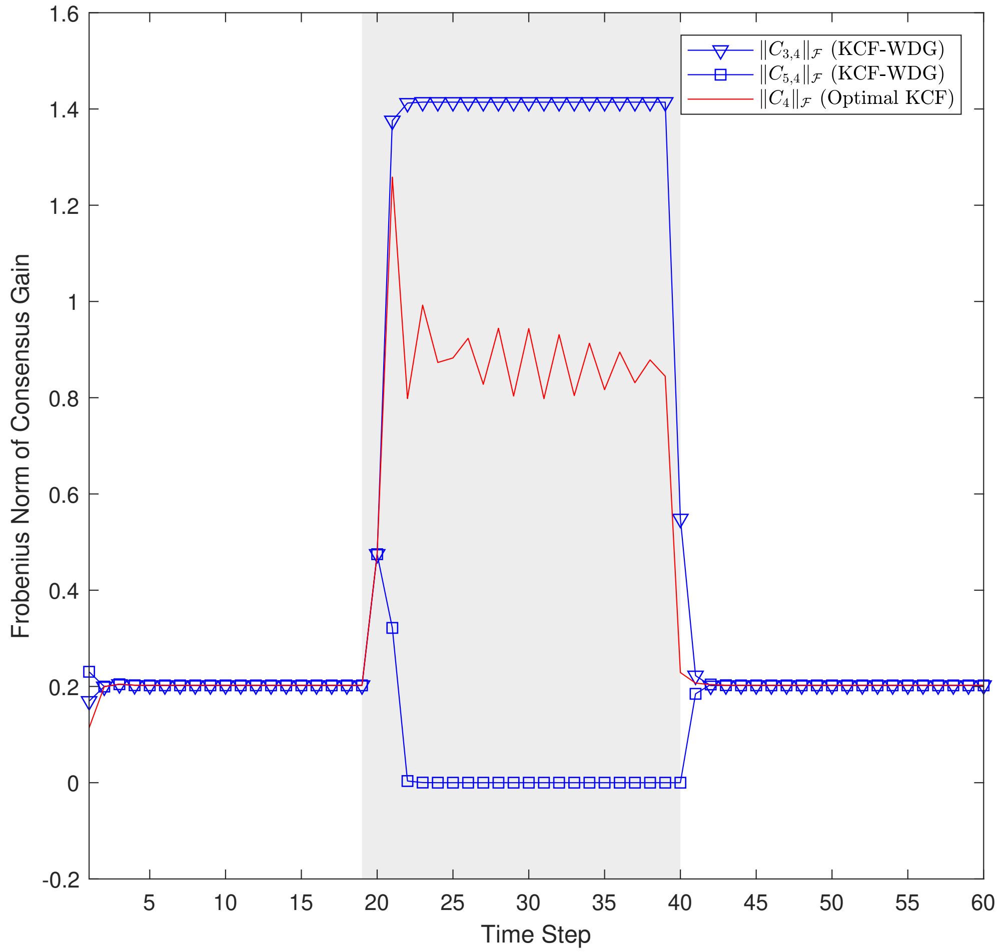


Figure 5.5.: **Scenario II** – Consensus gains computed at sensor 4 (averaged over 10,000 simulations), where the shaded region indicates the time steps where sensors 4, 5 and 6 cannot observe the target.

5.3 Scenario III : Limited Field of View Sensors

A common real world scenario where a part of the sensor network can be naïve, is in the case of a network of limited field of view (FOV) sensors, such as cameras. Camera measurements are typically instantaneous 2-dimensional projections of 3-dimensional objects in motion, and therefore a singular camera sensor has limited observability of the target. Moreover, cameras have a limited field of view, i.e., when the physical location of the target is outside a certain region (henceforth called the FOV) defined by the sensor specifications, the camera can no longer record measurements of the target in a reliable manner.

In this simulation, we consider a 4-dimensional target, following the dynamical model in (2.1), with the system matrices

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and $B = I_4$ and initial condition $x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$. The system noise covariance is

$$Q = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$

The resulting trajectory of the target starts at the origin and moves in a random direction away from it. The target can be thought of as an enemy agent attempting to escape from the simulation area, wherein the camera sensors attempt to estimate its position and velocity.

The sensor network observing the target has the cycle topology shown in Fig. 5.6, consisting of 7 sensors connected cyclically. The sensor network surrounds the target from all sides. Of these, each sensor has the observation model

$$H_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

which is a realistic model for sensors such as cameras, since cameras measurements are typically 2-dimensional image frames. Each sensor has a limited FOV which is triangular in shape, with the apex of this triangle is anchored at the camera (depicted in Fig. 5.7). The apex angle of this triangle is $80°$ and the height of the triangle is 200 units.

Sensor i belongs to the set $\mathcal{N}_{Naïve}$ if and only if the target is not contained in the FOV of sensor i. The sensor naïvety is imposed by varying the sensor noise covariance as

$$R_i = \begin{cases} 100 I_2, & i \in \mathcal{N}_{Naïve} \\ 10^5 I_2, & i \notin \mathcal{N}_{Naïve} \end{cases}$$

We run the simulation for 20 time steps. The rationale behind the simulation parameters is that the target starts from the origin and moves towards one of the sensors, which circumscribe the target. At the first time step, nearly all the sensors can observe the target, but towards the end of the simulation most sensors are oblivious of the target. As a result, it becomes imperative for the sensors to utilize the information received from their respective neighbors in an optimal manner, in order to accurately estimate the target.

Figures 5.8, 5.9 and 5.10 show the trajectory of the target for one such simulation, overlaid with the estimates of sensor 1 using the algorithms KCF, OKCF and OKCF-WDG, respectively. We choose to plot the estimates of sensor 1 in particular as it is on the far end of the network when compared to the final position of the target.

In the KCF algorithm (Fig. 5.8), the estimate of sensor 1 diverges from the true position of the target when the latter takes a sharp right turn. As the neighbors of sensor 1 are naïve as well, this sensor is unable to estimate the target over the course

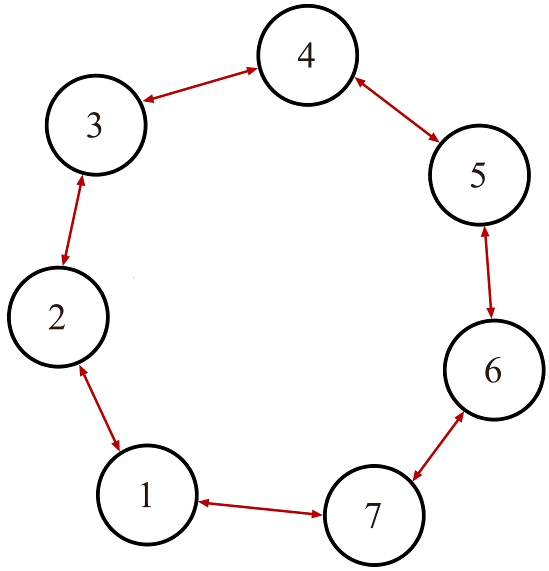


Figure 5.6.: **Scenario III** – Sensor network topology

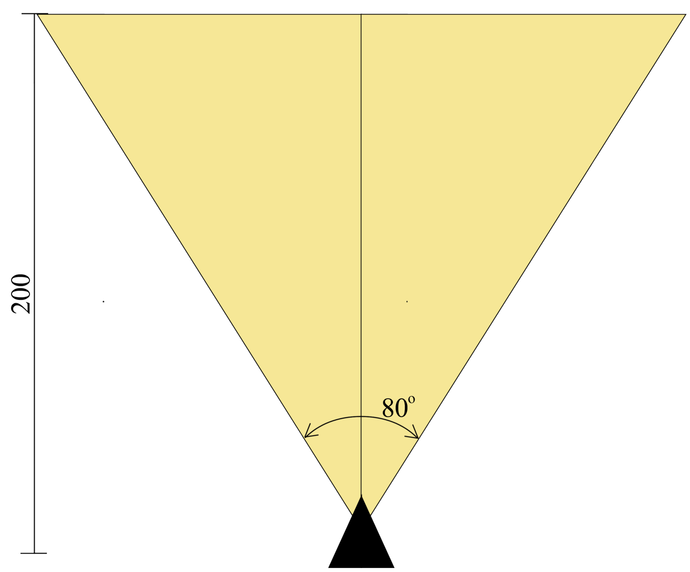


Figure 5.7.: **Scenario III** – Sensor specifications (shaded region represents its FOV)

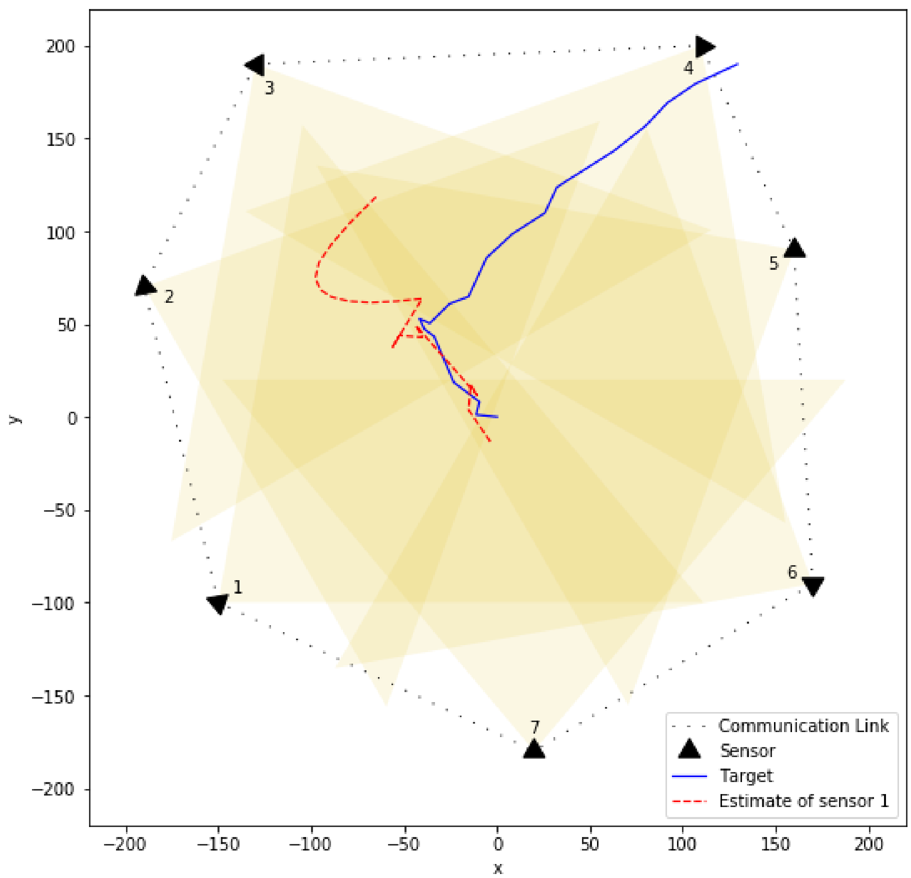


Figure 5.8.: **Scenario III** – Estimates of sensor using the **KCF** algorithm, simulated for 20 time steps

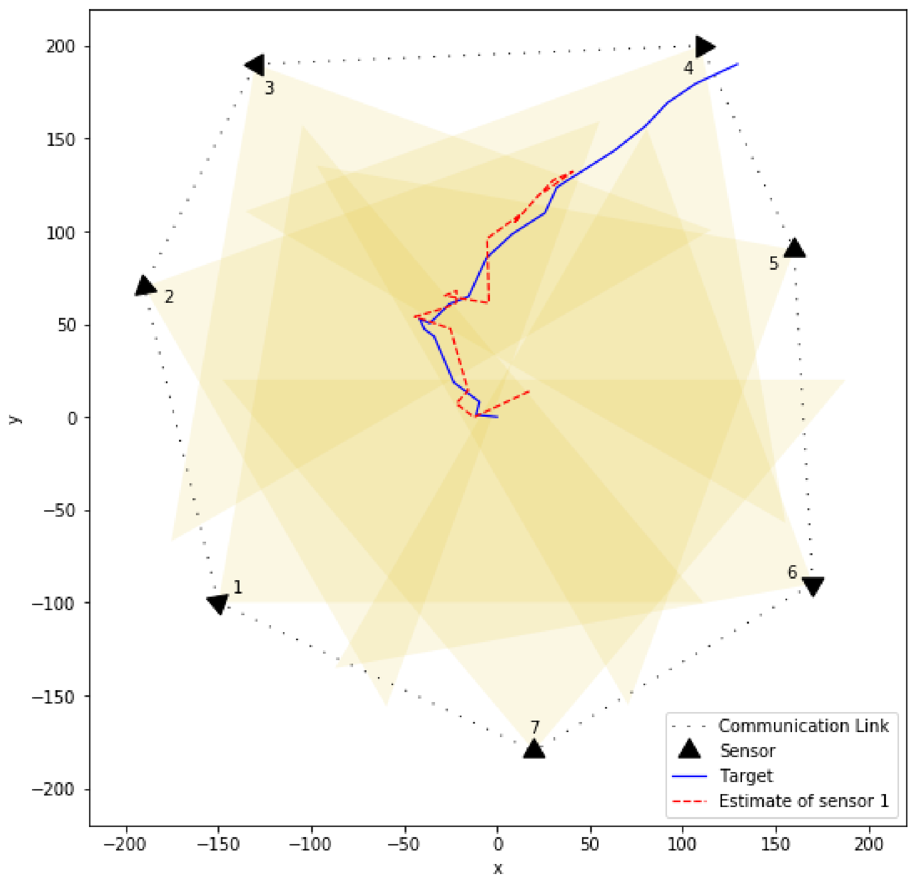


Figure 5.9.: **Scenario III** – Estimates of sensor using the **OKCF** algorithm, simulated for 20 time steps

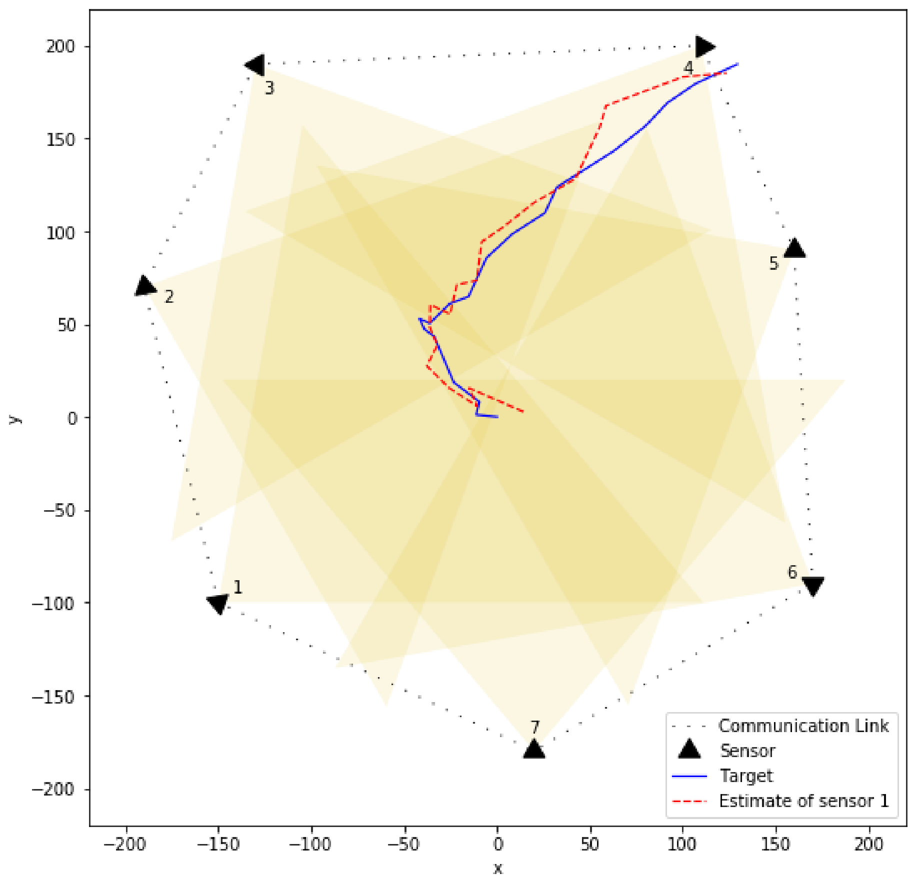


Figure 5.10.: **Scenario III** – Estimates of sensor using the **OKCF-WDG** algorithm, simulated for 20 time steps

of the simulation. Using the OKCF algorithm (Fig. 5.9), sensor 1 is able to estimate the state of the target better than KCF, but its estimate lags behind the true state of the target. This is because the propagation of the information within the network is slow on account of the algorithm utilizing unweighted consensus.

In contrast, the OKCF-WDG algorithm (Fig. 5.10) is able to estimate the target most accurately. The estimate of sensor 1 is able to keep up with the trajectory of the target even as a majority of the sensor network becomes naïve. It should be noted that similar trends were seen in other evaluations of this simulation as well.

To compare the average trend of estimation performance of each algorithm, we use the Monte Carlo method. Figures 5.11 and 5.12 show the mean squared estimation error at sensors 1 and 7 respectively, averaged over 10,000 Monte Carlo simulations. The simulations where the target escapes (all sensors become naïve) within 15 time steps were discarded. It is observed that the mean squared error of the KCF algorithm increases exponentially towards the end of the simulation. The mean squared error of the OKCF algorithm is 2 to 3 times that of the OKCF-WDG algorithm, which once again showcases the best performance of the three.

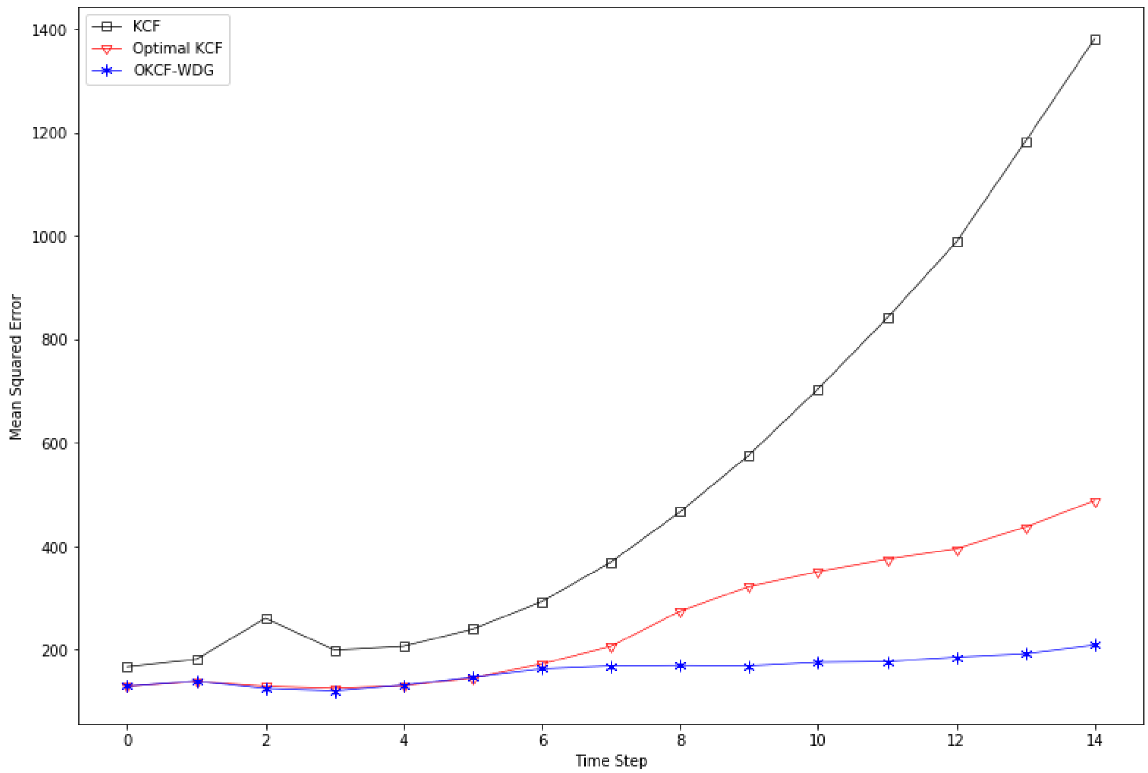


Figure 5.11.: **Scenario III** – Mean squared error of sensor 1, averaged over 10,000 simulations

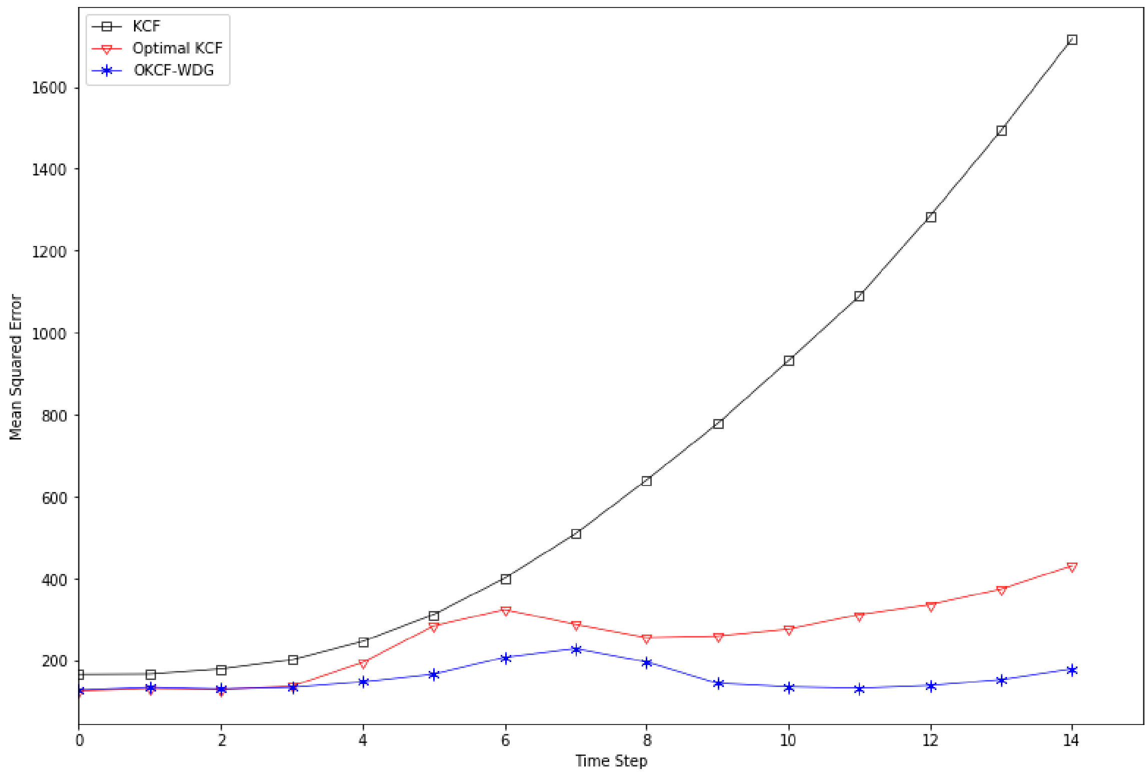


Figure 5.12.: **Scenario III** – Mean squared error of sensor 7, averaged over 10,000 simulations

6. CONCLUSIONS

In this thesis, we developed a novel distributed estimation algorithm that was motivated by the drawbacks of unweighted consensus algorithms such as the Kalman Consensus Filter (KCF). The defining characteristic of the proposed algorithm, which we call the Optimal Kalman Consensus Filter for Weighted Directed Graphs (OKCF-WDG), is that it uses a weighted consensus term at each sensor. This was accomplished by assigning different consensus gains to the estimates of each of the neighboring sensors. In contrast, KCF and several of its derived works utilize unweighted consensus.

The optimal estimation gains of the proposed algorithm were derived, which minimize the mean squared estimation error. The gains were obtained in a mathematically rigorous manner, without relying on approximations such as the ones made in several algorithms in distributed estimation literature. The performance of the algorithm was validated and compared against that of the sub-optimal and optimal KCF algorithms. It was noted that the OKCF-WDG algorithm exhibits significantly better estimation performance when some of the sensors in the network are naïve, i.e., cannot observe the target. A real world example where this is the case is limited field of view (FOV) sensors. A simulation scenario considering limited FOV sensors showcased the superior performance of OKCF-WDG.

Future work will focus on making the algorithm fully distributed. One way of achieving this is by implementing consensus sub-iterations at every estimation step, similar to what is done in the ICF protocol. However, since the OKCF-WDG algorithm uses the distributed information matrix, and not the centralized information matrix used in ICF, OKCF-WDG can be expected to require fewer consensus sub-iterations to achieve near-optimal estimation performance.

Another prospect for future work is that of the stability of the algorithm. It remains to be shown that the estimation error of OKCF-WDG remains bounded, which is something that could be studied using non-linear analyses such as Lyapunov stability theory.

REFERENCES

[1] Eric Foxlin, Dean Wormell, Thomas C Browne, and Michael Donfrancesco. Motion tracking system and method using camera and non-camera sensors, April 15 2014. US Patent 8,696,458.

[2] James D Spinhirne. Micro pulse lidar. *IEEE Transactions on Geoscience and Remote Sensing*, 31(1):48–55, 1993.

[3] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[4] Paul Baran. On distributed communications networks. *IEEE transactions on Communications Systems*, 12(1):1–9, 1964.

[5] Ashraf M Aziz. A simple and efficient suboptimal multilevel quantization approach in geographically distributed sensor systems. *Signal Processing*, 88(7):1698–1714, 2008.

[6] Wenling Li, Yingmin Jia, Junping Du, and Jun Zhang. Distributed consensus filtering for jump markov linear systems. *IET Control Theory & Applications*, 7(12):1659–1664, 2013.

[7] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185. ACM, 1999.

[8] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference*, volume 2, pages 1327–1332. IEEE, 2002.

[9] Seung-Min Oh and Eric Johnson. Relative motion estimation for vision-based formation flight using unscented kalman filter. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 6866, 2007.

[10] Reza Olfati-Saber and Parisa Jalalkamali. Coupled distributed estimation and control for mobile sensor networks. *IEEE Transactions on Automatic Control*, 57(10):2609–2614, 2012.

[11] Lang Hong. Centralized and distributed multisensor integration with uncertainties in communication networks. *IEEE transactions on aerospace and electronic systems*, 27(2):370–379, 1991.

[12] Jin Xue-Bo and Sun You-Xian. Optimal centralized state fusion estimation for multi-sensor system with correlated measurement noise. In *Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003.*, volume 1, pages 770–772. IEEE, 2003.

[13] Giorgio Battistelli and Luigi Chisci. Stability of consensus extended kalman filter for distributed state estimation. *Automatica*, 68:169–178, 2016.

[14] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000.

[15] Stergios I Roumeliotis and George A Bekey. Collective localization: A distributed kalman filter approach to localization of groups of mobile robots. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 3, pages 2958–2965. IEEE, 2000.

[16] Reza Olfati-Saber. Distributed kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 8179–8184. IEEE, 2005.

[17] Wei Ren, Randal W Beard, and Ella M Atkins. Information consensus in multivehicle cooperative control. *IEEE Control systems magazine*, 27(2):71–82, 2007.

[18] Luc Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on automatic control*, 50(2):169–182, 2005.

[19] Reza Olfati-Saber. Distributed kalman filtering for sensor networks. In *2007 46th IEEE Conference on Decision and Control*, pages 5492–5498. IEEE, 2007.

[20] Xuefeng Liu, Jiannong Cao, Wen-Zhan Song, Peng Guo, and Zongjian He. Distributed sensing for high-quality structural health monitoring using wsns. *IEEE Transactions on Parallel and Distributed Systems*, 26(3):738–747, 2015.

[21] Izudin Džafić, Rabih A Jabr, Indira Huseinagić, and Bikash C Pal. Multi-phase state estimation featuring industrial-grade distribution network models. *IEEE Transactions on Smart Grid*, 8(2):609–618, 2017.

[22] NS da Silva, A Simões Costa, KA Clements, and E Andreoli. Simultaneous estimation of state variables and network topology for power system real-time modeling. *Electric Power Systems Research*, 133:338–346, 2016.

[23] Yunpeng Hu, Wenshan Su, Lei Chen, and Yangang Liang. Cooperative space object tracking via universal kalman consensus filter. *Acta Astronautica*, 160:343–352, 2019.

[24] Ahmed Tashrif Kamal, Chong Ding, Bi Song, Jay A Farrell, and Amit K Roy-Chowdhury. A generalized kalman consensus filter for wide-area video networks. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7863–7869. IEEE, 2011.

[25] Ahmed T Kamal, Jay A Farrell, and Amit K Roy-Chowdhury. Information weighted consensus filters and their application in distributed camera networks. *IEEE Transactions on Automatic Control*, 58(12):3112–3125, 2013.

[26] Raj Deshmukh, Cheolhyeon Kwon, and Inseok Hwang. Optimal discrete-time kalman consensus filter. In *2017 American Control Conference (ACC)*, pages 5801–5806. IEEE, 2017.

[27] Ruggero Carli, Alessandro Chiuso, Luca Schenato, and Sandro Zampieri. Distributed kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in communications*, 26(4):622–633, 2008.

[28] Giorgio Battistelli, Luigi Chisci, Giovanni Mugnai, Alfonso Farina, and Antonio Graziano. Consensus-based linear and nonlinear filtering. *IEEE Transactions on Automatic Control*, 60(5):1410–1415, 2015.

[29] Shaocheng Wang, Wei Ren, and Zhongkui Li. Information-driven fully distributed kalman filter for sensor networks in presence of naive nodes. *arXiv preprint arXiv:1410.0411*, 2014.

[30] Reza Olfati-Saber. Kalman-consensus filter: Optimality, stability, and performance. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 7036–7042. IEEE, 2009.

[31] Jingbo Wu, Anja Elser, Shen Zeng, and Frank Allgöwer. Consensus-based distributed kalman-bucy filter for continuous-time systems. 49(22):321 – 326, 2016. 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2016.

[32] Shiraz Khan, Raj Deshmukh, and Inseok Hwang. Optimal kalman consensus filter for weighted directed graphs. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 7832–7837. IEEE, 2019.

[33] Minh Hoang Trinh, Chuong Van Nguyen, Young-Hun Lim, and Hyo-Sung Ahn. Matrix-weighted consensus and its applications. *Automatica*, 89:415–419, 2018.