

ספריות הטכניון *The Technion Libraries*

בית הספר ללימודי מוסמכים ע"ש ארווין וג'ואן ג'ייקובס
Irwin and Joan Jacobs Graduate School

©

All rights reserved to the author

This work, in whole or in part, may not be copied (in any media), printed, translated, stored in a retrieval system, transmitted via the internet or other electronic means, except for "fair use" of brief quotations for academic instruction, criticism, or research purposes only. Commercial use of this material is completely prohibited.

©

כל הזכויות שמורות למחבר/ת

אין להעתיק (במדיה כלשהי), להדפיס, לתרגם, לאחסן במאגר מידע, להפיץ באינטרנט, חיבור זה או כל חלק ממנו, למעט "שימוש הוגן" בקטעים קצרים מן החיבור למטרות לימוד, הוראה, ביקורת או מחקר. שימוש מסחרי בחומר הכלול בחיבור זה אסור בהחלט.

Consensus Kalman Filtering: Filter Design and Event- Triggering

Aviv Priel

Consensus Kalman Filtering: Filter Design and Event- Triggering

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace Engineering

Aviv Priel

Submitted to the Senate
of the Technion — Israel Institute of Technology
Adar 5782 Haifa February 2022

This research was carried out under the supervision of Prof. Daniel Zelazo, in the Faculty of Aerospace Engineering.

Some results in this thesis have been published as articles by the author and research collaborators in conferences and journals during the course of the author's research period, the most up-to-date versions of which being:

Aviv Priel and Daniel Zelazo. An improved distributed consensus Kalman filter design approach. In *2021 60th IEEE conference on Decision and Control*. IEEE, 2021.

Acknowledgements

Foremost, I would like to thank my partner in life - my dear wife which stepped forward when I needed some peace of mind to promote my research.

Next, I wish to express my admiration and gratitude to my advisor, Prof. Daniel Zelazo, for his stubborn support and for believing in me. His words were a beacon of light which guided me through vast planes of theory and academic procedures. He inspired me in ways that will stick with me for the rest of my professional and academic career.

Besides my advisor, I would like to thank the rest of my exam committee: Prof. Leonid Mirkin and Prof. Yaakov Oshman, for their time, their meaningful comments, and for the interesting conversation we had.

I also acknowledge the generous financial help of the Technion - Israel Institute of Technology.

Finally, I wish to acknowledge the generous financial help of the Li Ka Shing Fellowships.

Contents

List of Figures

Abstract	1
Abbreviations and Notations	3
1 Introduction	5
1.1 Literature Review	6
1.1.1 Consensus Kalman Filter (CKF)	6
1.1.2 Event-Triggered Consensus Kalman Filter (ETCKF)	7
1.2 Thesis Contribution	7
1.3 Thesis Organization	9
2 Preliminaries	11
2.1 The Discrete-Time Kalman Filter	11
2.1.1 Problem Statement	11
2.1.2 The Kalman Estimator	12
2.1.3 Discrete-Time Kalman Filter Algorithm	13
2.2 Graph Theory and Consensus Algorithms	14
2.2.1 Basic Definitions	14
2.2.2 Discrete-Time Consensus Algorithm	16
2.3 Event-Triggered Estimation	17
2.3.1 Classification	18
3 Consensus Kalman Filtering	21
3.1 Problem Setup	21
3.2 Consensus Kalman Filter Update Equation	22
3.2.1 Optimal Consensus Kalman Filter	22
3.2.2 Sub-Optimal Consensus Kalman Filter	25
3.3 An Improved Consensus Gain Selection	27
3.3.1 Centralized Consensus Gain Determination	27
3.3.2 Decentralized Consensus Gain Determination	31

3.4	Simulation Results	34
4	Event-Triggered Consensus Kalman Filter	43
4.1	Problem Setup	43
4.2	Event-Triggered Consensus Kalman Estimator	44
4.2.1	Event-Triggered Condition for a Centralized Consensus Gain	45
4.2.2	Event-Triggered Condition for a Decentralized Consensus Gain	49
4.3	Stability with Partial Non-Observability	53
4.4	Simulation Results	57
5	Conclusion and Open Questions	65
5.1	Conclusion	65
5.2	Future Work	66
	Hebrew Abstract	i

List of Figures

2.1	Recursive Kalman state estimation process.	13
2.2	Directed (right) and undirected (left) graphs.	15
2.3	Disconnected (right) and connected (left) graphs.	15
2.4	Trajectories of the consensus protocol, (2.17), for a network of 5 agents.	17
2.5	Event-triggering mechanism.	18
3.1	DCKE of the i th agent.	22
3.2	Two-hop neighborhood of node v_i	25
3.3	Consensus factor (3.12) proposed in [33] as a function of time for example presented in Section 3.4.	27
3.4	DCKE structure for the i th agent - centralized consensus gain architecture.	31
3.5	DCKE structure for the i th agent - decentralized consensus gain architecture.	32
3.6	A sensor network of 20 agents randomly positioned.	35
3.7	Standard deviation of the agents' state estimation for both axes, comparison between 6 distributed state estimators over 100 Monte-Carlo runs for a homogeneous sensing model.	36
3.8	Root mean squared error, comparison between 7 state estimators over 100 Monte-Carlo runs for a homogeneous sensing model.	37
3.9	Standard deviation of the agents' state estimation for both axes, comparison between 6 state estimators over 100 Monte-Carlo runs for a heterogeneous sensing model.	38
3.10	Root mean squared error, comparison between 7 state estimators over 100 Monte-Carlo runs for a heterogeneous sensing model.	38
3.11	Trajectory of the true state and the agents' mean estimate utilizing SOCKF2 (a) and DSOCKF2 (b) for a heterogeneous sensing model.	39
3.12	Local MSE, comparison between agents with minimum NCLKF MSE, maximum SOCKF2 MSE and maximum DSOCKF2 MSE for a single run with a heterogeneous sensing model.	39
3.13	Communication graph at time steps (a) 1-49, (b) 50-149, and (c) 150-300.	40
3.14	Sum of all agents mean squared error with two graph switches (at step 50 and at step 150), comparison between 3 state estimators for a single run with the heterogeneous sensing model.	40

4.1	DETCKE of the i^{th} agent.	45
4.2	DETCKE structure for the i th agent - centralized consensus gain architecture.	48
4.3	DETCKE structure for the i th agent - decentralized consensus gain architecture.	49
4.4	Available v.s instantaneous communication topology where agents broadcast information based on an event triggering mechanism.	50
4.5	Measurement model where sensing agents can only sense the process when it is in range.	54
4.6	A sensor network of 20 agents randomly positioned.	58
4.7	Root mean squared error, comparison between 5 state estimators over 100 Monte-Carlo runs with a homogeneous sensing model.	59
4.8	Total events per agent for a 400 step simulation - a comparison between 3 event triggered estimators and the CKF with a homogeneous sensing model.	59
4.9	Trajectory of the true state and the agents' mean estimate utilizing ETCKF1 (a) and ETDCKF (b) for the homogeneous model.	60
4.10	Root mean squared error, comparison between 5 state estimators over 100 Monte-Carlo runs with a heterogeneous sensing scheme.	61
4.11	Total events per agent for a 400 step simulation - a comparison between 3 event triggered estimators and the CKF with a heterogeneous sensing scheme.	61
4.12	Trajectory of the true state and the agents' mean estimate utilizing ETCKF1 (a) and ETDCKF (b) for the heterogeneous model.	62
4.13	Communication graph at time steps (a) 1-49, (b) 50-149, and (c) 150-400.	62
4.14	Sum of all agents mean squared error for 2 mid-run graph switch (at step 50 and at step 150), comparison between 3 state estimators for a single run.	63
4.15	Available communication graph for a sensing network with 20 agents.	63
4.16	Observing and non-observing agents for an 85 meters out of range limitation at $k = 50$ (a) and $k = 150$ (b).	64
4.17	Partial observability scenario results.	64
5.1	Cooperative estimation network designers toolbox.	66

Abstract

Sensor networks comprise a group of agents equipped with sensing devices and communicating capabilities in order to solve the common task of cooperative estimation of a detectable physical process. In this framework, each agent in the system activates, in a distributed fashion, an estimator which relies on local measurements fused with the estimates from other agents in the network. A recently developed tool to solve this problem is the introduction of a consensus-based term fused with a classical Kalman state estimator structure, known as the *consensus Kalman filter*. Our contribution begins with proposing a method based on semi-definite programming to compute a centralized consensus gain term, leading to improved performance of the estimator over existing solutions found in the literature. We also propose a decentralized consensus gain, for networks with homogeneous observation models, that can be computed by each agent and relies only on local network properties (the number of neighboring agents).

We further extend our research to tackle the important aspect of reducing energy (communication) consumption in network applications. To do so, we utilize an event-triggering mechanism in which communication is permitted only if certain conditions are met. The main analytical challenge in these estimator structures is the design of the consensus gain term and an event-triggered condition that ensures stability of the estimation error dynamics. In this direction, our contribution continues with proposing both a centralized and a decentralized consensus gain along with a tailored event-triggered condition. We show that these event-triggered estimators out-perform the standard non-cooperative local Kalman filter. Finally we introduce an event-triggered consensus Kalman estimator which can cope with real-life scenarios where some agents may have intermittent or absent observations. We provide numerical simulations to demonstrate the effectiveness of our results compared to existing solutions in the literature.

Abbreviations and Notations

\mathbb{R}^n	: set of real n -dimensional vectors
$\mathbb{R}^{n \times m}$: set of real $n \times m$ matrices
\mathbb{R}^+	: set of positive real numbers
$\mathbb{E}(x)$: mean value of x
$\text{diag}\{M^i\}_{i=1}^n$: block diagonal $nd \times nd$ matrix where the i^{th} block is $M^i \in \mathbb{R}^{d \times d}$
$[M]_{ij}$: ij -entry of the matrix M
$\lambda_{\max}(M)$: maximal eigenvalue of the matrix M
$\lambda_{\min}(M)$: minimal eigenvalue of the matrix M
$\sigma_{\max}(M)$: maximal singular value of the matrix M
$\sigma_{\min}(M)$: minimal singular value of the matrix M
$\text{tr}(M)$: trace of matrix M
$ M _F$: the Frobenius norm of the matrix M
WSN	: wireless sensor networks
DETE	: distributed event-triggered estimation
CKF	: consensus Kalman filter
MSE	: mean squared error
NCLKF	: non-cooperative local Kalman filter
DCKE	: distributed consensus Kalman estimator
ETCKF	: event-triggered consensus Kalman filter
SoD	: send on delta
MAS	: multi-agent systems
ETM	: event-triggered mechanism
ETF	: event-triggered function
ETC	: event-triggered condition
SOCKF	: sub-optimal consensus Kalman filter
ZOH	: zero order hold

Chapter 1

Introduction

Sensor networks comprise a group of agents equipped with sensing devices and communicating capabilities in order to solve some common task such as cooperative sensing and estimation of a detectable physical process. This complex problem has been a major subject of interest in various research communities due to its wide range of applications including agriculture [30], oceanographic monitoring [2], security and surveillance [51, 36], health monitoring [27] and space research [43].

One of the fundamental challenges in sensor networks deal with cooperative estimation of some globally observable process [1, 7]. In this scenario, each agent in the system activates, in a distributed fashion, an estimator which relies on local measurements of the process fused with the estimates from other agents in the network. The motivation to utilize neighboring information is to exploit the information that exists in the sensors network to improve the performance of an individual agent, as well as the overall network estimation accuracy. As a whole, the networked system aims to globally converge to the true process state while considering constraints such as computational loads, the amount of data shared, and the overall system performance.

A recently developed tool to solve this problem is the introduction of a consensus-based term fused with a classical state estimator structure [10]. This provides a mechanism for accounting for neighboring information. For example, the consensus H_∞ estimator is discussed in [44], the distributed particle filtering as presented in [16], and a consensus Kalman filter was formulated in [32, 4]. In this work we shall deal with the consensus Kalman filter, for which the estimator is composed of a classic Kalman estimator structure fused with a consensus-based term.

Due to recent advance in manufacturing capabilities, the implementation of large scaled wireless sensor networks (WSN) became more feasible. The unavoidable increase in scale in WSN is well discussed in [9]. Considering cooperative estimation, this increase in scale has instigated the necessity to discuss constraints such as network band-

width [38] and global with local energy consumption (computational loads, transmission power, etc...) [55]. Essentially, the trade-off between the estimator performance and the data transmission required to obtain it, came into mind. For some researchers [20, 28], the estimator cost function is no longer solely based on the estimation accuracy, but introduces a new component aimed at capturing the penalty associated to transmissions energy consumption.

One of the mechanisms found in order to cope with these constraints is referred to as *distributed event-triggered estimation* (DETE). In this scheme, each agent in the system retrieves or transmits information to its neighbors only when some rule is violated. The rule comprises a set of conditions which must be locally examined at each iteration step. Violation of the rule will trigger an event and information will be shared. Between events, the agents will run a Kalman based estimator that relies only on locally obtained information. The system aims to globally converge to the true process state.

1.1 Literature Review

In this section we shall review works dealing with the consensus Kalman filter, with and without an event triggering mechanism, to cover the basic foundation for our work and to provide some motivation for our contributions.

1.1.1 Consensus Kalman Filter (CKF)

The *consensus Kalman filter* was first proposed by Olfati-Saber in [31], where he showed that the problem of distributed Kalman filtering in sensor networks can be solved by activating a consensus based filter on the state estimation (and inverse covariance) of each sensor. Since then, one could witness increasing interest in the CKF as new works are occupying different aspects of the filter. In [8], the consensus Kalman estimator proposed in [32] was adapted, and they then derived the solution for both Kalman and consensus gains that will minimize the local mean squared error (MSE). Additionally, the authors of [8] have compared simulation results with the sub-optimal solution suggested in [33]. The work [46] utilized the same sub-optimal solution to derive a consensus extended Kalman filter in order to solve a spacecraft network relative motion estimation problem. The authors in [6] made another variation on the sub-optimal consensus Kalman filter discussed in [33] to solve the extended problem of networks with agent that have limited or null measurement capabilities.

In these recent papers, to our knowledge, no comparison has been made between sub-optimal consensus Kalman filter and the *non-cooperative local Kalman filter* (NCLKF), where each sensor implements a Kalman filter without any exchange of information from

other sensors in the network. Furthermore, the selected consensus gain derived in [33] might obtain small values rendering the consensus term contribution insignificant. In this case the estimator behaves more like a NCLKF without agents reaching agreement on their estimates.

1.1.2 Event-Triggered Consensus Kalman Filter (ETCKF)

Traditionally, state estimators such as the discrete Kalman filter [18], assumes that measurements are acquired at every time step. This mechanism was outdated as constraints and restrictions on data transmission and bandwidth arose, especially while discussing wireless sensor networks [3]. Researchers began seeking for efficient ways to deliver measurements and estimates over the network. In this direction, much work has been done to investigate intelligent methods for data sampling according to some rule-set, see [23, 12]. In this way, fewer data samples are dispatched in order to hopefully achieve a similar estimator performance. These methods are widely referred to as *event-triggered estimation* mechanisms.

Specifically, the consensus Kalman filter combined with an event-triggering mechanism is a subject of increasing interest in recent years. For example, in [21], the *send on delta* (SoD) rule is discussed, where each agent transmits its local estimates to its neighbours only if the difference between the most recent transmitted estimate and the current estimate exceeds some threshold. The author of [54] expanded this research while addressing data transmission from a sensor to its peer estimator, where the estimator to estimator communication is discussed. For this scenario, the event trigger condition is based on measurements only. In [22], the issue of intermittent observations was tackled by introducing a binary variable to the update equations.

It should be noted that, to our knowledge, no comparison has been made between this approach and NCLKF, where each sensor implements a Kalman filter without any exchange of information from other sensors in the network. Furthermore, the event-triggering mechanism might be overly sparse, rendering the consensus term contribution insignificant. In this case the estimator behaves more like a NCLKF without agents reaching agreement on their estimates. Additionally the aspect of time varying communication regime has not been formally introduced or investigated.

1.2 Thesis Contribution

In this section we outline the contribution of our work to the world of cooperative estimation. Specifically we seek to solve the cooperative estimation of a linear discrete time

process with a wireless sensor network. We aim to do so in two fields of interest: distributed consensus Kalman filtering, and distributed event-triggered estimation.

We begin with deriving the consensus Kalman filter proposed by Olfati-Saber in [33], where we point out some issues with respect to suggested consensus gains found in the literature. Next, we propose an alternative consensus gain which is computed utilizing convex optimization techniques. We use the Lyapunov theorem in order to extract constraints for our convex optimization problem. With this proposed solution, we derive for each time step an upper bound for the consensus factor for which stability of the estimator is ensured. By using this upper bound we ensure that the consensus term, encouraging the agreement of estimates between neighboring agents, plays a nontrivial role in the estimator dynamics. We demonstrate through simulation examples the superiority in performance over the NCLKF and other consensus gain found in the literature. We also show the proposed estimator is mean-squared error Lyapunov stable. The aforementioned consensus gain is computed in a centralised manner since global network matrices are required, therefore our contribution proceeds with proposing a decentralized consensus gain, for networks with a homogeneous sensing model, which is based on local network properties and thus can be implemented in systems with switching or time-varying communication networks. Once more, superiority in performance over the NCLKF and others are presented through simulations results. Our contribution in this area is summarised in the following list:

- i) extracting an upper bound on the consensus gain factor using convex optimization and proposing a centralized consensus factor based on the extracted upper bound;
- ii) proposing a decentralized consensus gain which is suitable for time-varying communication networks for a homogeneous observation architecture.

For the event-triggered consensus Kalman filter, our contribution continues with proposing a centralized consensus gain along with an event-triggering condition for which the stability of the error dynamics is ensured. Once more the consensus gain is computed centrally. Next, we propose an event-triggered mechanism which corresponds to a proposed decentralized consensus gain for networks with a homogeneous sensing model. With this solution, our event trigger scheme is robust to time-varying communication topology.

Additionally, we construct a strategy to cope with real-life scenarios such as sensors that may only function intermittently, for example due to occlusion between the sensor and the process, or malfunction. In these scenarios, some agents' observability of the process can be intermittent or absent and therefore we use consensus based mechanism to ensure the stability of the distributed mechanism. Finally, we provide extended comparisons through numerical simulation between different schemes including the NCLKF where the energy consumption v.s performance trade-off is discussed. Our contribution

is summarised in the following list:

- iv) proposing an event-triggered mechanism together with a centralized and decentralized consensus gain;
- v) proposing an event trigger mechanism, for a homogeneous observation architecture, which is robust to temporary non-observability of some agents ;
- vi) providing numerical evidence for the superiority of our proposed solution in both areas.

1.3 Thesis Organization

This work is organized as follows. In Chapter 2, we provide an introduction to three fields of research which are the pillars to this work: the classic Kalman filter, graph theory (with a specific orientation to the consensus algorithms), and for event-triggered estimation. In Chapter 3 we discuss the fusion of a consensus algorithm and a Kalman filter to solve the cooperative estimation problem. In Chapter 4, we apply an event-triggered mechanism on the consensus Kalman filter to reduce communication loads and in Chapter 5 our work is concluded with some suggested guidelines for future research.

Chapter 2

Preliminaries

In this chapter we shall introduce the research topics which will serve as the building blocks for this thesis.

2.1 The Discrete-Time Kalman Filter

The discrete-time Kalman filter was first introduced by R.E Kalman in [19] as a recursive solution for the discrete linear data filtering problem. This recursive algorithm was proven compatible for many application, including trajectory estimation and control of spacecraft in the late 60's. Ever since then, the Kalman filter became an essential component of modern control systems. For more extensive introduction to the Kalman filter, the reader is encouraged to see [19, 25, 47].

2.1.1 Problem Statement

Consider the following discrete-time linear system,

$$\begin{aligned}x_{k+1} &= Ax_k + Su_k + w_k \\z_k &= Hx_k + v_k,\end{aligned}\tag{2.1}$$

where k is the time step, $x_k \in \mathbb{R}^n$ is the state vector, $u_k \in \mathbb{R}^q$ is a control input, $z_k \in \mathbb{R}^m$ is an observation (measurement), $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^m$ are, respectively, the process and measurement noises. Additionally $A \in \mathbb{R}^{n \times n}$ represents the state matrix, $S \in \mathbb{R}^{n \times q}$ is the control input matrix and $H \in \mathbb{R}^{m \times n}$ is the observation matrix.

The process and measurement noises are assumed to be independent of each other

$(\mathbb{E}[v_k^T w_l] = 0 \forall k, l)$, white, and with normal probability distributions such that

$$w_k \sim N(0, Q) \quad (2.2)$$

$$v_k \sim N(0, R). \quad (2.3)$$

Recall that $x \sim N(\mu, \Sigma)$ means that the random vector x has a mean value $\mathbb{E}(x) = \mu$, a covariance matrix $\mathbb{E}((x - \mathbb{E}(x))(x - \mathbb{E}(x))^T) = \Sigma$ and a probability density function

$$p(x) = \frac{1}{(2\pi)^n |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}.$$

Given Q, R, A, S, H, z_k, u_k , and some initial estimate \bar{x}_0 , we seek to estimate the process x_k that minimizes the mean-squared estimation error,

$$J_k = \mathbb{E} \left[(x_k - \mathbb{E}(x_k))^T (x_k - \mathbb{E}(x_k)) \right]. \quad (2.4)$$

2.1.2 The Kalman Estimator

To solve the problem presented in the previous subsection, Kalman used the orthogonality principle (see [18, 42]) to construct a *posteriori* state estimation which is a linear combination of an *a priori* estimate along with a weighted difference between a measurement and its predicted value,

$$\hat{x}_k = \bar{x}_k + K_k (z_k - H\bar{x}_k), \quad (2.5)$$

where \hat{x}_k is the *a posteriori* estimate, \bar{x}_k is the *a priori* estimate, and K_k is the weight given to the measurement prediction error, known as the *innovation process*, and can be "tuned" to achieve a particular performance. Additionally, K_k is also known as the *Kalman gain*. For a high gain, more weight is applied to the most recent measurements, rendering the estimator to rely on them more.

The *a priori* state estimation \bar{x}_k is the noise-free state propagation in the absence of new information about the state (for example, a new measurement),

$$\bar{x}_{k+1} = A\hat{x}_k + Su_k. \quad (2.6)$$

As illustrated in Figure 2.1, the recursive state estimation process takes the form of a feedback loop where the state is being propagated while a feedback in the form of the measurement prediction is obtained. In other words the Kalman estimation process can be divided into two groups: the time update (state prediction) and the measurement update (state estimation). This separation shall become distinct as we proceed with

the formulation of the Kalman filter update equations.

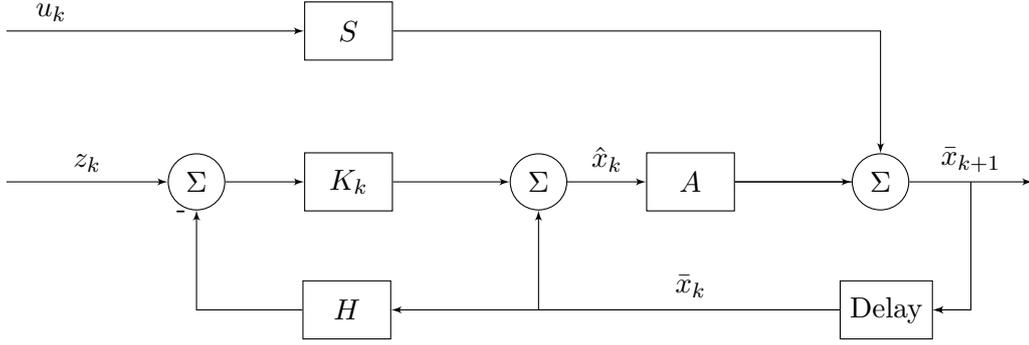


Figure 2.1: Recursive Kalman state estimation process.

2.1.3 Discrete-Time Kalman Filter Algorithm

Now that the Kalman estimator structure is in place, it is left to decide on the optimal Kalman gain which will minimize the mean-squared estimation error (MSEE). To do so, Kalman first defined the estimator error dynamics,

$$\begin{aligned}\eta_k &= \hat{x}_k - x_k = \bar{x}_k + K_k(z_k - H\bar{x}_k) - x_k \\ &= (I - K_k H)\bar{\eta}_k + K_k v_k \\ \bar{\eta}_{k+1} &= \bar{x}_{k+1} - x_{k+1} = A\eta_k - w_k,\end{aligned}\tag{2.7}$$

next he defined the *a priori* error covariance estimate, $\bar{P}_k = \mathbb{E}[\bar{\eta}_k \bar{\eta}_k^T]$, and *posteriori* error covariance estimate, $\hat{P}_k = \mathbb{E}[\eta_k \eta_k^T]$.

It follows that the error covariances are,

$$\begin{aligned}\hat{P}_k &= F_k \bar{P}_k F_k^T + K_k R K_k^T \\ \bar{P}_{k+1} &= A \hat{P}_k A^T + Q,\end{aligned}\tag{2.8}$$

where $F_k = I - K_k H$. Minimizing the MSE is equivalent to minimizing the trace of the error covariance, hence, the optimal Kalman gain solves the following equation,

$$\frac{\partial \text{tr} \left(\mathbb{E}[\eta_k \eta_k^T] \right)}{\partial K_k} = 0.\tag{2.9}$$

The following properties of matrix calculus are used,

$$\frac{\partial \text{tr}(Y X^T)}{\partial X} = Y, \quad \frac{\partial \text{tr}(X Y)}{\partial X} = Y^T, \quad \frac{\partial \text{tr}(X Y X^T)}{\partial X} = X(Y + Y^T).\tag{2.10}$$

Thus, the optimality condition can be obtained as

$$\frac{\partial \text{tr}(\hat{P}_k)}{\partial K_k} = -2\bar{P}_k H^T + 2K_k H \bar{P}_k H^T + 2K_k R = 0, \quad (2.11)$$

and the optimal Kalman gain is

$$K_k = \bar{P}_k H^T (H \bar{P}_k H^T + R)^{-1}. \quad (2.12)$$

It is straight forward to validate that (2.12) represent a global minimum since the second derivative of the error covariance trace is positive definite. The discrete Kalman filter update equation are obtained:

$$\text{KF} : \begin{cases} \mathbf{Estimation} \\ K_k = \bar{P}_k H^T (R + H \bar{P}_k H^T)^{-1} \\ \hat{P}_k = F_k \bar{P}_k F_k^T + K_k R K_k^T \\ \hat{x}_k = \bar{x}_k + K_k (z_k - H \bar{x}_k) \\ \mathbf{Prediction} \\ \bar{x}_{k+1} = A \hat{x}_k + S u_k \\ \bar{P}_{k+1} = A \hat{P}_k A^T + Q. \end{cases} \quad (2.13)$$

2.2 Graph Theory and Consensus Algorithms

In this introductory section we provide some basic graph theory background to the material that we present more formally in later chapters.

2.2.1 Basic Definitions

A graph \mathcal{G} is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a finite vertex set $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ with N vertices, and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. A graph is called *directed* if its edges have a direction which is specified with the ordering of the vertices pairs, i.e., if $(v, u) \in \mathcal{E}$ it does not necessarily means that $(u, v) \in \mathcal{E}$ for some $u, v \in \mathcal{V}$ (see Fig. 2.2(b)). An *undirected graph* is one where all edges are bi-directional, i.e., $(v, u) \in \mathcal{E}$ implies $(u, v) \in \mathcal{E} \forall v, u \in \mathcal{V}$ (see Fig. 2.2(a)). A *path* is a sequence of vertices where each vertex share an edge with its predecessor vertex and its successor vertex. A *directed path* is a path where the ordering of the vertices corresponds with the edges direction. In undirected graphs, the neighborhood (set of neighbors) of the vertex v is defined as all the vertices which share an edge with v , $\mathcal{N}_v = \{u \in \mathcal{V} \mid (u, v) \in \mathcal{E}\}$. The degree of a vertex, $d_v = |\mathcal{N}_v|$, is the number of edges incident to the vertex v , and the degree matrix

D is a diagonal matrix where $[D]_{ii} = d_i$. For graphs where each vertex is assigned with at least one edge, the degree matrix is invertible. In directed graphs, the in-degree of a vertex, $d_v^{in} = |\{u \in \mathcal{V} \mid (u, v) \in \mathcal{E}\}|$, is the number of edges which are directed to the vertex and the out-degree of a vertex, $d_v^{out} = |\{u \in \mathcal{V} \mid (v, u) \in \mathcal{E}\}|$, is the number edges which are directed from the vertex.

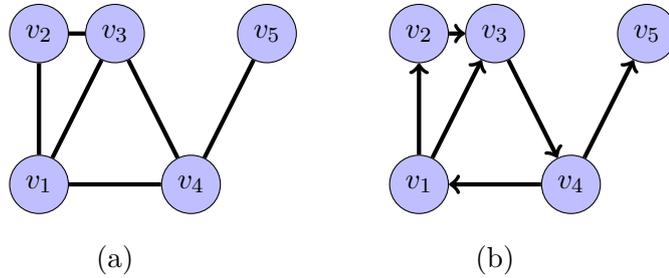


Figure 2.2: Directed (right) and undirected (left) graphs.

A directed graph is called *disconnected* if for its undirected version (replacing all directed edges with undirected edges) there are at least 2 vertices with no connecting path between them (see Fig. 2.3(b)). A graph is *connected* if it is not disconnected (see Fig. 2.3(a)). A directed graph is called *weakly connected* if its undirected version is a connected graph. A directed graph is called *strongly connected* if there is a directed path between every pair of vertices.

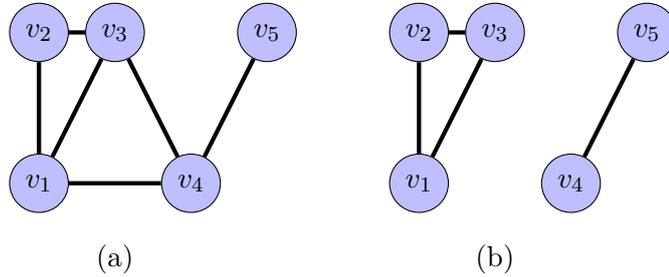


Figure 2.3: Disconnected (right) and connected (left) graphs.

Any undirected graph can be represented by the Laplacian matrix, $L \in \mathbb{R}^{N \times N}$ [5], where,

$$L_{i,j} = \begin{cases} d_i & i = j \\ -1 & (i, j) \in \mathcal{E}, i \neq j \\ 0 & \text{otherwise} \end{cases} \quad (2.14)$$

Note that the Laplacian has an eigenvalue 0, corresponding to the eigenvector, $\mathbf{1}$, the vector of all ones (i.e., $\mathbf{1}_i = 1$ for all i). For directed graphs we can distinguish between the in-degree Laplacian and the out-degree Laplacian which are not necessarily symmetric.

2.2.2 Discrete-Time Consensus Algorithm

Multi-agent systems (MAS) refers to a group of agent aimed at obtaining some common goal. These type of systems have been a hot research topic due to their wide range of application in various fields such as vehicle formations [37, 40], rendezvous problems [24], and coordinated decision making [41]. The key element in applying such coordination is by sharing information between agents. In a distributed MAS, information is shared locally between neighboring agents, and then propagated via inter-agent information exchange to the rest of the system. With this idea emerged the requirement to find a protocol which will drive the system towards agreement on some common quantity of interest.

Consensus protocols, as formulated in [35], refers to the process of achieving global agreement between all acting agents upon a certain state. In this protocol, we consider a network of N agents which interact over some communication network described by a graph \mathcal{G} . Each agent has a state $x^i \in \mathbb{R}$ with initial value x_0^i . The consensus (sometimes referred to as agreement) protocols aim to drive each agent to the same state or trajectory, i.e.,

$$\lim_{k \rightarrow \infty} \|x_k^i - x_k^j\| = 0, \quad \forall i, j \in \mathcal{V}. \quad (2.15)$$

For example, consider a collection of n agents modeled as discrete-time integrators,

$$x_{k+1}^i = x_k^i + u_k^i.$$

Each agent then implements the following control,

$$u_k^i = \epsilon \sum_{j \in \mathcal{N}_i} (x_k^j - x_k^i),$$

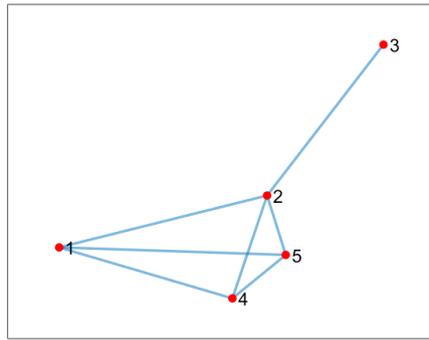
yielding the closed-loop model

$$x_{k+1}^i = x_k^i + \epsilon \sum_{j \in \mathcal{N}_i} (x_k^j - x_k^i), \quad (2.16)$$

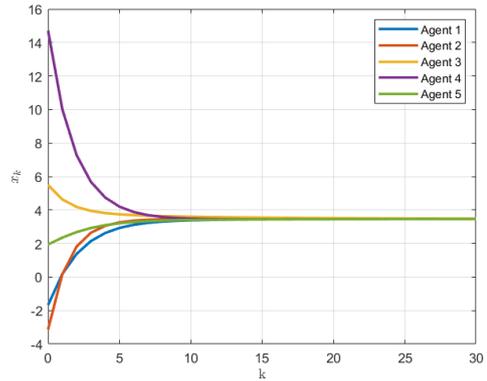
for some $\epsilon \in \mathbb{R}$ chosen to ensure stability. The state-space dynamics can now be constructed as

$$\mathbf{x}_{k+1} = (I - \epsilon L) \mathbf{x}_k, \quad (2.17)$$

where L is the Laplacian matrix (2.14) and $\mathbf{x}_k = [x_k^1, x_k^2, \dots, x_k^N]^T$. Assuming that $I - \epsilon L$ is stable, this system has an equilibrium at $\mathbf{x}_k = \bar{x} \mathbf{1}$, meaning that agreement is obtained. An example of a network of 5 agents running the consensus protocol (2.17) with $\epsilon = 0.1$ is shown in Fig. 2.4.



(a) The interaction graph.



(b) Agent trajectories.

Figure 2.4: Trajectories of the consensus protocol, (2.17), for a network of 5 agents.

2.3 Event-Triggered Estimation

In the past decade, the art of economical computation via event triggering and the requirement for robustly stable filters formed the highly niche research field of *event-triggered estimation*. The motivation for utilizing event-triggered protocols lies within bandwidth restriction and/or power supply limitation. To demonstrate this idea, [12] illustrated an example in which a network of underwater wireless sensors can not comply with consecutive transmission protocols in order to perform cooperative estimation. This is due to the low bit rate delivery and limited resource budget. Indeed, one may question the efficiency of transmitting two consecutive sampled data points with little fluctuation between them.

For these constrained systems, one may consider implementing an *event-triggered mechanism* (ETM) in order to regulate sufficiently the data transmission or the sampling action. In this sense, if the mechanism is properly designed, the sampling is conducted in an intelligent manner so that new data will flow only when the “necessity” arises. In that way, utilization of computational and/or transmission resources is reduced and performance of the system estimation is not drastically compromised.

Whether referring to a single sensor-estimator system or cooperative estimation utilizing WSN, the following scheme represents a typical event-triggered mechanism (see Fig. 2.5): The sampler is responsible for sampling and holding the discrete data signal θ_k . The sampled signal is denoted by θ_{τ_s} where τ_s , for $s = 0, 1, 2, \dots$, represent the sampling instance sub-sequence with $\tau_0 = 0$. Note that the value of the sampled signal can be updated only if an event is triggered, while in between events it will hold the same value. The sampling is enabled only when there is a violation of the event-triggered condition, which consists of locally available knowledge denoted by Φ_k , the signal θ_k , and the most recent sampled signal, θ_{τ_s} . Evaluated at each time step, the condition serves as the tactic to inspect whether the system requires another sampling or may

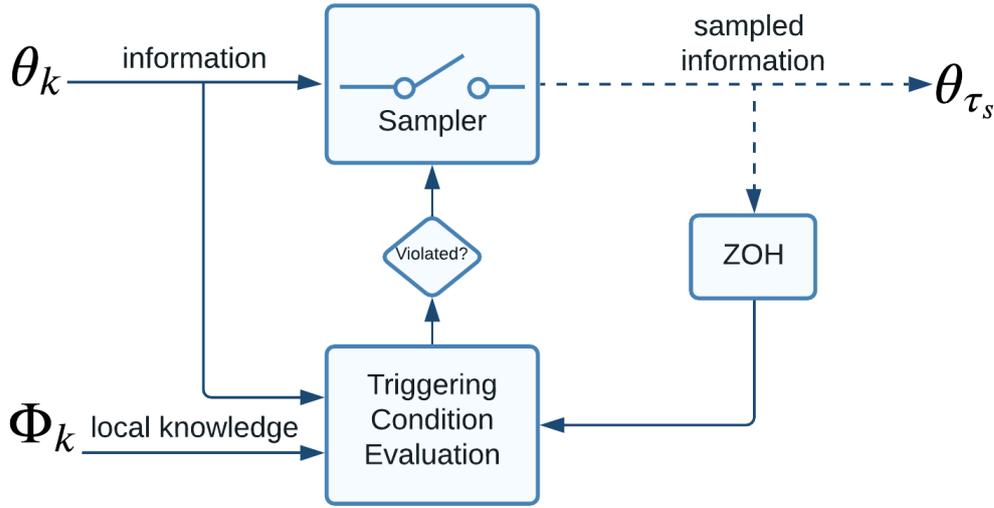


Figure 2.5: Event-triggering mechanism.

proceed with the latest sampled data.

The triggering mechanism is constructed in the following manner:

$$\tau_{s+1} = \inf_k \{k > \tau_s | f(\theta_k, \theta_{\tau_s}, \Phi_k) > \delta_k\} \quad (2.18)$$

where δ_k is some threshold, $f(\cdot)$ is the *event triggering function* (ETF), and the rule $f(\theta_k, \theta_{\tau_s}, \Phi_k) \leq \delta_k$ is referred to as the *event triggering condition* (ETC). There are several of common methods to construct the ETF as will be discussed in the following subsection.

2.3.1 Classification

In this subsection we shall discuss the different types of common event-triggered mechanisms. For these types, the ETF usually utilizes an error between a quantity and its sampled quantity in the event-free period as presented below:

- i) *Constant Threshold Based Event-Triggered Mechanisms* : In this case, $\delta_k = \delta$ where $\delta \geq 0$ represents a constant threshold. This mechanism is commonly referred as *send-on-delta* (SoD) mechanism. For example, in [54], the SoD event-triggered mechanisms is applied on sensor networks where the sensor to estimator channel is discussed. In this case, the data signal θ_k is the sensor measurement and the ETC is constructed as

$$(\theta_k - \theta_{\tau_s})^T (\theta_k - \theta_{\tau_s}) < \delta. \quad (2.19)$$

Once violated, the most recent measurement shall be transmitted to the estimator. Additionally, [54] occupied the estimator-to-estimator channel with the same mechanism (2.19), only the signal to be sampled is now the inter-agent state estimation. In this case, if the condition is violated an estimator will transmit its most recent estimation to its neighboring estimators.

- ii) *Measurement/Estimate Based Event-Triggered Mechanisms* : In this case, the event-triggering threshold from (2.18), obtains the following general form: $\delta_k = \delta(\theta_k)$. In many works, the measurement/estimate dependent threshold, $\delta(\theta_k)$, incorporates a weighting factor on the latest measurements/ estimate such that an event shall be triggered when the error between the most recent measurement/estimate and the weighted measurement/estimate from previous event, exceeds some specified value.

For example, in [15], the event-triggered distributed H_∞ consensus filtering problem was approached using the following event-triggered condition:

$$(\theta_k - \theta_{\tau_s})^T \Omega (\theta_k - \theta_{\tau_s}) < \sigma \theta_k^T \Omega \theta_k, \quad (2.20)$$

where θ is the state estimation, $\sigma \in (0, 1]$ is some weighing factor and Ω is some positive-definite matrix.

- iii) *Transmitted Measurement/Estimate Based Event-Triggered Mechanisms* : In this case, the event-triggering threshold from (2.18) takes the form $\delta_k = \delta(\theta_{\tau_s})$. In other words the threshold to trigger an event is a function of the measurement/estimate at the time of the most recent event (last transmitted value).

For example, in [53], the event-triggered mechanism, based on the latest transmitted measurement, is applied to solve the distributed set membership filter for a class of discrete time-varying systems in the presence of unknown but bounded noises. The proposed ETC for each system is:

$$(\theta_k - \theta_{\tau_s})^T \Omega^{-1} (\theta_k - \theta_{\tau_s}) < \sigma (\theta_{\tau_s})^T \theta_{\tau_s}, \quad (2.21)$$

where θ is the state measurement, Ω is some positive definite matrix and $\sigma \in \mathbb{R}^+$ is some weighing factor.

For more information with respect to recent developments in the realm of event-triggered estimation, the reader is encourage to see [12, 11, 23].

Chapter 3

Consensus Kalman Filtering

In Chapter 2 we introduced the Kalman filter and the consensus algorithm. In this chapter we shall formulate and discuss the fusion of a consensus term in a local Kalman estimator as a tool to account for neighboring information of a given process. Moreover, we will follow a similar design approach for the determination of a consensus gain proposed by [33]. We propose a new consensus gain design that will lead to an improvement in performance.

3.1 Problem Setup

We consider a network comprising N interacting sensor agents where the interaction topology can be described by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ indicates which agents can exchange information with each other. Each agent observes a linear discrete-time stochastic process described by the dynamics

$$\mathcal{P} : x_{k+1} = Ax_k + Bw_k, \quad (3.1)$$

where $x_k \in \mathbb{R}^n$ is the state vector and w_k is an additive white Gaussian noise such that $\mathbb{E} [w_k(w_l)^T] = Q\delta_{kl}$, where δ_{kl} is the Dirac Delta function.

Each agent is capable of measuring the process state using the observation model

$$z_k^i = H^i x_k + v_k^i, \quad (3.2)$$

where $z_k^i \in \mathbb{R}^{m^i}$ is the measurement obtained by agent i , $H^i \in \mathbb{R}^{m^i \times n}$ is the observation matrix, and $v_k^i \in \mathbb{R}^{m^i}$ is a measurement noise assumed to also be additive white Gaussian noise with $\mathbb{E} [v_k^i(v_l^i)^T] = R^i\delta_{kl}$. Additionally we assume that $R^i \in \mathbb{R}^{m^i \times m^i}$ is invertible and that (A, H^i) make an observable pair for every agent such that the noiseless NCLKF is asymptotically stable.

The distributed *consensus Kalman* estimator (DCKE) was first proposed by [34] and is constructed as

$$\hat{x}_k^i = \bar{x}_k^i + K_k^i \left(z_k^i - H^i \bar{x}_k^i \right) + C_k^i \sum_{j \in \mathcal{N}_i} \left(\bar{x}_k^j - \bar{x}_k^i \right), \quad (3.3)$$

where K^i and C^i are the Kalman and consensus gains of the i th agent, respectively, and \hat{x}^i and \bar{x}^i are the *posteriori* and *a priori* state estimate of the i th agent, respectively. The consensus Kalman estimator (3.3) is composed of a classic Kalman estimator term and a consensus term based on neighbors estimates, as illustrated in Fig. 3.1.

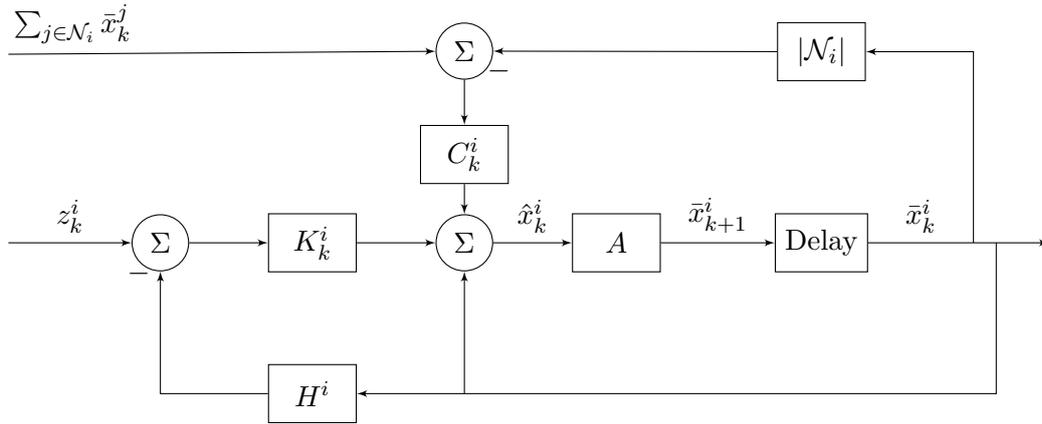


Figure 3.1: DCKE of the i th agent.

3.2 Consensus Kalman Filter Update Equation

In this section we shall follow in the footsteps of [8] to derive the optimal consensus Kalman filter update equations. Next, we shall discuss some troubling aspects with respect to the latter solution which will then follow by presenting a sub-optimal solution which was suggested by Olfati-Saber in [33]. This will serve as the grounds for our contribution in the following sections.

3.2.1 Optimal Consensus Kalman Filter

We begin our discussion by the problem we wish to solve.

Problem 3.1 (Optimal consensus Kalman gains). *Consider N agents interacting over a connected graph \mathcal{G} where each observes the process (3.1) with observation model (3.2) and utilizes an estimator type (3.3). Find a set of local optimal gains, K_k^i and C_k^i , that will minimize the local mean-squared estimation error.*

To solve this problem we first construct the error dynamics,

$$\begin{aligned}
 \eta_k^i &= \hat{x}_k^i - x_k = \bar{x}_k^i + K_k^i \left(Hx_k + v_k^i - H^i \bar{x}_k^i \right) + C_k^i \sum_{j \in \mathcal{N}_i} \left(\bar{x}_k^j - \bar{x}_k^i \right) - x_k \\
 &= F_k^i \bar{\eta}_k^i + C_k^i \sum_{j \in \mathcal{N}_i} \left(\bar{x}_k^j - \bar{x}_k^i \right) + K_k^i v_k^i \\
 &= F_k^i \bar{\eta}_k^i + C_k^i \sum_{j \in \mathcal{N}_i} \left(\bar{\eta}_k^j - \bar{\eta}_k^i \right) + K_k^i v_k^i \\
 \bar{\eta}_{k+1}^i &= \bar{x}_{k+1}^i - x_{k+1} = A \eta_k^i - B w_k,
 \end{aligned}$$

where $F_k^i = I - K_k^i H^i$. Next we compute the inter-agent correlation term,

$$\begin{aligned}
 \mathbb{E} \left[\eta_k^i (\eta_k^r)^T \right] &= F_k^i \bar{P}_k^{i,r} (F_k^r)^T + F_k^i \sum_{s \in \mathcal{N}_r} \left(\bar{P}_k^{i,s} - \bar{P}_k^{i,r} \right)^T (C_k^r)^T \\
 &\quad + C_k^i \sum_{j \in \mathcal{N}_i} \left(\bar{P}_k^{j,r} - \bar{P}_k^{i,r} \right) (F_k^r)^T + K_k^i R^{i,r} (K_k^r)^T + C_k^i D_k^{i,r} (C_k^r)^T,
 \end{aligned}$$

where

$$\begin{aligned}
 \bar{P}^{i,s} &= \mathbb{E} \left[\bar{\eta}^i (\bar{\eta}^s)^T \right], & \bar{P}^{i,r} &= \mathbb{E} \left[\bar{\eta}^i (\bar{\eta}^r)^T \right] \\
 \bar{P}^{j,s} &= \mathbb{E} \left[\bar{\eta}^j (\bar{\eta}^s)^T \right], & \bar{P}^{j,r} &= \mathbb{E} \left[\bar{\eta}^j (\bar{\eta}^r)^T \right] \\
 R^{i,r} &= \mathbb{E} \left[v^i (v^r)^T \right],
 \end{aligned}$$

and $D^{i,r} = \sum_{j \in \mathcal{N}_i} \sum_{s \in \mathcal{N}_r} \left(\bar{P}^{j,s} - \bar{P}^{i,s} - \bar{P}^{j,r} + \bar{P}^{i,r} \right)$. Note that $R^{i,r} = \mathbb{E} \left[v_k^i (v_k^r)^T \right] = 0$ if $i \neq r$. It follows that the internal agent's error covariance is

$$\begin{aligned}
 \mathbb{E} \left[\eta_k^i (\eta_k^i)^T \right] &= F_k^i \bar{P}_k^i (F_k^i)^T + F_k^i \sum_{j \in \mathcal{N}_i} \left(\bar{P}_k^{i,j} - \bar{P}_k^i \right)^T (C_k^i)^T \\
 &\quad + C_k^i \sum_{j \in \mathcal{N}_i} \left(\bar{P}_k^{j,i} - \bar{P}_k^i \right) (F_k^i)^T + K_k^i R^i (K_k^i)^T + C_k^i D_k^i (C_k^i)^T.
 \end{aligned}$$

We shall find the optimal local Kalman gain by deriving the local error covariance trace by the Kalman gain and equating to zero,

$$\begin{aligned}
 \frac{\partial \text{tr} \left(\mathbb{E} \left[\eta_k^i (\eta_k^i)^T \right] \right)}{\partial K_k^i} &= -2P_k^i (H^i)^T + 2K_k^i H^i \bar{P}_k^i (H^i)^T - 2C_k^i \sum_{j \in \mathcal{N}_i} \left(\bar{P}_k^{j,i} - P_k^i \right) (H^i)^T \\
 &\quad + 2K_k^i R^i = 0.
 \end{aligned} \tag{3.4}$$

Therefore the optimal Kalman gain is

$$K_k^i = \left(P_k^i (H^i)^T + C_k^i \sum_{j \in \mathcal{N}_i} \left(\bar{P}_k^{j,i} - \bar{P}_k^i \right) (H^i)^T \right) \left(R^i + H^i \bar{P}_k^i (H^i)^T \right)^{-1}. \tag{3.5}$$

The update rule is as follow:

$$\text{CKF : } \left\{ \begin{array}{l}
 \textbf{Estimation} \\
 K_k^i = \left(\bar{P}_k^i (H^i)^T + C_k^i \sum_{j \in \mathcal{N}_i} (\bar{P}_k^{j,i} - \bar{P}_k^i) (H^i)^T \right) (R^i + H^i \bar{P}_k^i (H^i)^T)^{-1} \\
 \hat{P}_k^{i,r} = F_k^i \bar{P}_k^{i,r} (F_k^r)^T + F_k^i \sum_{s \in \mathcal{N}_r} (\bar{P}_k^{i,s} - \bar{P}_k^{i,r})^T (C_k^r)^T \\
 \quad + C_k^i \sum_{j \in \mathcal{N}_i} (\bar{P}_k^{j,r} - \bar{P}_k^{i,r}) (F_k^r)^T + K_k^i R^{i,r} (K_k^r)^T + C_k^i D_k^{i,r} (C_k^r)^T \\
 \hat{x}_k^i = \bar{x}_k^i + K_k^i (z_k^i - H^i \bar{x}_k^i) + C_k^i \sum_{j \in \mathcal{N}_i} (\bar{x}_k^j - \bar{x}_k^i) \\
 \textbf{Prediction} \\
 \bar{x}_{k+1}^i = A \hat{x}_k^i \\
 \bar{P}_{k+1}^{i,r} = A \hat{P}_k^{i,r} A^T + B Q B^T
 \end{array} \right. \quad (3.6)$$

We proceed with finding the optimal consensus gain. To do so, first we shall express the error covariance as a function of C_k^i only:

$$\begin{aligned}
 \mathbb{E} \left[\eta_k^i (\eta_k^i)^T \right] &= \bar{P}_k^i + \bar{P}_k^i \Omega_k^i H^i \bar{P}_k^i (H^i)^T (\Omega_k^i)^T \bar{P}_k^{iT} + C_k^i \Gamma_k^i \Omega_k^i H^i \bar{P}_k^i (H^i)^T (\Omega_k^i)^T (\Gamma_k^i)^T (C_k^i)^T \\
 &\quad + \bar{P}_k^i \Omega_k^i H^i \bar{P}_k^i (H^i)^T (\Omega_k^i)^T (\Gamma_k^i)^T (C_k^i)^T + C_k^i \Gamma_k^i \Omega_k^i H^i \bar{P}_k^i (H^i)^T (\Omega_k^i)^T (\bar{P}_k^i)^T \\
 &\quad - \bar{P}_k^i (H^i)^T (\Omega_k^i)^T (\bar{P}_k^i)^T - \bar{P}_k^i (H^i)^T (\Omega_k^i)^T (\Gamma_k^i)^T (C_k^i)^T - \bar{P}_k^i \Omega_k^i H^i \bar{P}_k^i \\
 &\quad - C_k^i \Gamma_k^i \Omega_k^i H^i \bar{P}_k^i + (\Gamma_k^i)^T C_k^{iT} + C_k^i \Gamma_k^i - \bar{P}_k^i \Omega_k^i H^i (\Gamma_k^i)^T C_k^{iT} - C_k^i \Gamma_k^i \Omega_k^i H^i (\Gamma_k^i)^T C_k^{iT} \\
 &\quad - C_k^i \Gamma_k^i (H^i)^T (\Omega_k^i)^T \bar{P}_k^i - C_k^i \Gamma_k^i (H^i)^T (\Omega_k^i)^T (\Gamma_k^i)^T (C_k^i)^T \\
 &\quad + \bar{P}_k^i \Omega_k^i R^i (\Omega_k^i)^T \bar{P}_k^{iT} + C_k^i \Gamma_k^i \Omega_k^i R^i (\Omega_k^i)^T (\Gamma_k^i)^T (C_k^i)^T + \bar{P}_k^i \Omega_k^i R^i (\Omega_k^i)^T (\Gamma_k^i)^T (C_k^i)^T \\
 &\quad + C_k^i \Gamma_k^i \Omega_k^i R^i (\Omega_k^i)^T (\bar{P}_k^i)^T + C_k^i D_k^i (C_k^i)^T,
 \end{aligned}$$

where $\Gamma_k^i = \sum_{j \in \mathcal{N}_i} (\bar{P}_k^{j,i} - \bar{P}_k^i)$ and $\Omega_k^i = (H^i)^T (R^i + H^i \bar{P}_k^i (H^i)^T)^{-1}$. For optimality, we use the matrix calculus (see (2.10)) to derive and solve the following equation:

$$\begin{aligned}
 \frac{\partial \text{tr} \left(\mathbb{E} \left[\eta_k^i (\eta_k^i)^T \right] \right)}{\partial C_k^i} &= 2C_k^i \Gamma_k^i \Omega_k^i H^i \bar{P}_k^i (H^i)^T (\Omega_k^i)^T (\Gamma_k^i)^T - 2C_k^i \Gamma_k^i \Omega_k^i H^i (\Gamma_k^i)^T \\
 &\quad + 2\bar{P}_k^i \Omega_k^i H^i \bar{P}_k^i (H^i)^T (\Omega_k^i)^T (\Gamma_k^i)^T - 2\bar{P}_k^i (H^i)^T (\Omega_k^i)^T (\Gamma_k^i)^T \\
 &\quad - 2C_k^i \Gamma_k^i (H^i)^T (\Omega_k^i)^T (\Gamma_k^i)^T + 2C_k^i \Gamma_k^i \Omega_k^i R^i (\Omega_k^i)^T (\Gamma_k^i)^T \\
 &\quad + 2\bar{P}_k^i \Omega_k^i R^i (\Omega_k^i)^T (\Gamma_k^i)^T + 2(\Gamma_k^i)^T - 2C_k^i \Gamma_k^i \Omega_k^i H^i (\Gamma_k^i)^T \\
 &\quad - 2\bar{P}_k^i \Omega_k^i H^i (\Gamma_k^i)^T + 2C_k^i D_k^i = 0,
 \end{aligned} \quad (3.7)$$

We note that

$$\Omega_k^i H^i \bar{P}_k^i (H^i)^T (\Omega_k^i)^T (\Gamma_k^i)^T + \Omega_k^i R^i (\Omega_k^i)^T (\Gamma_k^i)^T = \Omega_k^i H^i (\Gamma_k^i)^T,$$

and that $\Omega_k^i H^i = (H^i)^T (\Omega_k^i)^T$. Hence, the following is obtained:

$$\frac{\partial \text{tr} \left(\mathbb{E} \left[\eta_k^i (\eta_k^i)^T \right] \right)}{\partial C_k^i} = 2(\Gamma_k^i)^T - 2\bar{P}_k^i \Omega_k^i H^i (\Gamma_k^i)^T - 2C_k^i \Gamma_k^i (H^i)^T (\Omega_k^i)^T (\Gamma_k^i)^T + 2C_k^i D_k^i = 0. \quad (3.8)$$

Therefore, the optimal consensus gain is

$$C_k^i = \left((\Gamma_k^i)^T - \bar{P}_k^i \Omega_k^i H^i (\Gamma_k^i)^T \right) \left(\Gamma_k^i (H^i)^T (\Omega_k^i)^T (\Gamma_k^i)^T - D_k^i \right)^{-1}. \quad (3.9)$$

To this end we wish to emphasize two main points. Firstly, the consensus gain (3.9) may be ill-conditioned as no guarantees are made regarding the structure of the matrix $\Gamma_k^i (H^i)^T (\Omega_k^i)^T (\Gamma_k^i)^T - D_k^i$. Secondly, as will be discussed in the following subsection, to update the matrix D_k^i , one requires additional communication channels to a two-hop neighborhood (see Fig. 3.2). This may place unwanted burdens on the communication bandwidth. Hence, one may consider implementing a more “economical” solution at the expense of performance.

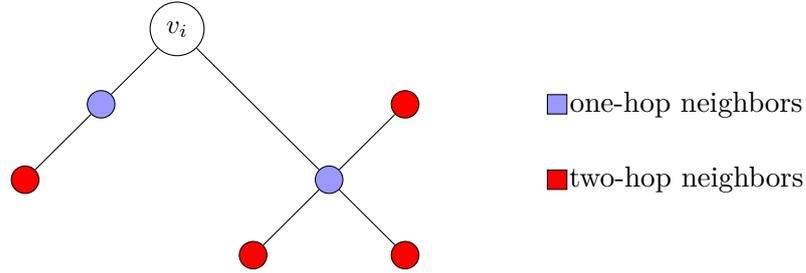


Figure 3.2: Two-hop neighborhood of node v_i .

3.2.2 Sub-Optimal Consensus Kalman Filter

Olfati-Saber in [33] constructed the distributed optimal Kalman filter (3.6) without explicitly computing the optimal consensus gain. Instead, he discussed the flaws that exist in the update equations optimal form. Specifically the necessity to retrieve two-hop neighborhood information in order to update the cross correlation term was emphasized. Not only that a two-hop neighborhood data flow is hard to implement, it can also lead to computational overloads. For example, in a complete graph this would mean that each agent would retrieve $N(N + 1)$ cross correlation terms at each step. The latter served as the motivation to construct a sub-optimal distributed consensus

Kalman filter (SOCKF) which solves the following problem:

Problem 3.2 (Sub-Optimal consensus Kalman gains). *Consider N agents interacting over a connected graph \mathcal{G} where each observes the process (3.1) with observation model (3.2) and utilizes an estimator type (3.3). Using a one-hop based information exchange, find a set of local gains, K_k^i and C_k^i , that will ensure the stability of the estimation error.*

Olfati-Saber proposed the following update equations which utilizes only (one-hop) neighboring state estimates and discards the consensus terms from the error covariance and Kalman gain equations in (3.6).

$$\left\{ \begin{array}{l} \mathbf{Estimation} \\ \begin{aligned} K_k^i &= P_k^i (H^i)^T (R^i + H^i \bar{P}_k^i (H^i)^T)^{-1} \\ \hat{P}_k^i &= F_k^i \bar{P}_k^i (F_k^i)^T + K_k^i R^i K_k^{iT} \\ \hat{x}_k^i &= \bar{x}_k^i + K_k^i (z_k^i - H^i \bar{x}_k^i) + C_k^i \sum_{j \in \mathcal{N}_i} (\bar{x}_k^j - \bar{x}_k^i) \end{aligned} \\ \mathbf{Prediction} \\ \begin{aligned} \bar{x}_{k+1}^i &= A \hat{x}_k^i \\ \bar{P}_{k+1}^i &= A \hat{P}_k^i A^T + B Q B^T, \end{aligned} \end{array} \right. \quad (3.10)$$

where $F_k^i = I - K_k^i H^i$. The omission of the consensus terms from the Kalman gain and error covariance update equation is justified with the assumption that the consensus gain is relatively small. It was shown in [33] that (3.10) has stable estimator dynamics with an appropriate consensus gain selection, However, it should be noted that one must be careful while selecting a small consensus gain since this might lead the consensus component in the DCKE to be negligible.

For example the gain from [33] is given as

$$C_k^i = \gamma_k P_k^i (F_k^i)^{-T} \quad (3.11)$$

with the consensus factor

$$\gamma_k = \sqrt{\frac{\lambda_{\min}(\Psi_k)}{\lambda_{\max}((L \otimes A) Y_k (L \otimes A))}}, \quad (3.12)$$

where $\Psi_k = \text{diag}\{(\hat{P}_{k-1}^i)^{-1} - A^T (F_k^i)^T (\hat{P}_k^i)^{-1} F_k^i A\}_{i=1}^N$ and $Y_k = \text{diag}\{(F_k^i)^{-1} (\hat{P}_k^i)^{-1} (F_k^i)^{-T}\}_{i=1}^N$. Although this consensus factor ensures the stability of the estimator, $\lambda_{\min}(\Psi_k)$ can obtain small values prior to convergence. This will result in a negligible consensus factor and thus the cooperative nature of the estimator may be overstated. This idea is captured in Fig. 3.3, where the consensus factor (3.12) is depicted. The consensus factor was calculated according to the numerical example provided in Section 3.4. As

can be seen, γ_k obtains very small values in a short period of time. This will cause the inter-agent contribution in the estimation to be insignificant. Further discussion will be provided in Section 3.4. This motivated us to search for a consensus gain that will provide a meaningful network-level contribution.

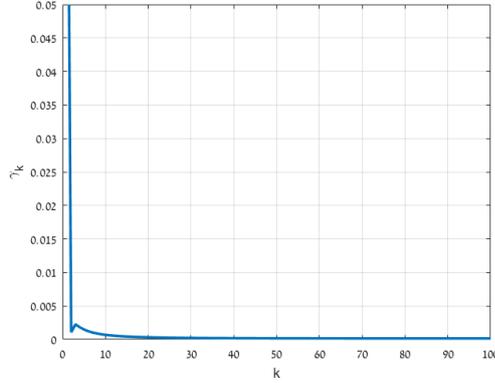


Figure 3.3: Consensus factor (3.12) proposed in [33] as a function of time for example presented in Section 3.4.

3.3 An Improved Consensus Gain Selection

In this section we explore both centralized and decentralized approaches for designing the consensus gain term C_k^i in (3.10).

3.3.1 Centralized Consensus Gain Determination

We propose a new consensus gain for the SOCKF update scheme (3.10). We aim to extract the maximal consensus gain in a manner that will ensure the stability of the local estimation error (and thus, for the sum of all errors as well).

Theorem 3.3 (DCKE Stability). *Consider a group of N agents interacting over a connected graph \mathcal{G} where each observes the process (3.1) with observation model (3.2). The noiseless estimation error with the Kalman consensus filter (3.10) and the choice of consensus gain $C_k^i = \gamma_k P_k^i (F_k^i)^{-T}$ is asymptotically stable for any $\gamma_k \in [0, \gamma_k^*] \forall k$, where γ_k^* can be obtained as the maximum value for which*

$$\begin{aligned} \mathcal{K}_k(\gamma_k) = & \text{diag} \left\{ (\hat{P}_{k-1}^i)^{-1} - A^T (F_k^i)^T (\hat{P}_k^i)^{-1} F_k^i A \right\}_{i=1}^N + 2\gamma_k (L \otimes A^T A) \\ & - \gamma_k^2 (L \otimes A)^T \text{diag} \{ (F_k^i)^{-1} \hat{P}_k^i (F_k^i)^{-T} \}_{i=1}^N (L \otimes A), \end{aligned} \quad (3.13)$$

is positive semi-definite, and can be found using semi-definite programming.

Proof. The proof for this theorem follows the same line as presented in [33] with an additional section to establish the range of consensus gains γ_k . First we choose a quadratic Lyapunov function and show that for $\gamma_k = 0$, the Lyapunov function is monotonically decreasing. We then prove that there must be some γ_k^* such that for any $\gamma_k \in [0, \gamma_k^*]$, the Lyapunov function is monotonically decreasing. Finally we show that γ_k^* can be found using semi-definite programming (SDP).

Let $\eta_k = \hat{x}_k - x_k$ and $\bar{\eta}_k = \bar{x}_k - x_k$ be the estimation and prediction errors, respectively. The noiseless error dynamics are

$$\begin{aligned}\eta_k^i &= \underbrace{(I - K_k^i H^i)}_{F_k^i} \bar{\eta}_k^i + C_k^i \sum_{j \in \mathcal{N}_j} (\bar{\eta}_k^j - \bar{\eta}_k^i) \\ \bar{\eta}_{k+1}^i &= A \eta_k^i.\end{aligned}$$

Consider now the following Lyapunov function,

$$V_k = \sum_{i=1}^N (\eta_k^i)^T (\hat{P}_k^i)^{-1} \eta_k^i. \quad (3.14)$$

The Lyapunov step difference function along the system trajectories is

$$\begin{aligned}\delta V_k &= V_k - V_{k-1} \\ &= \sum_{i=1}^N (\eta_k^i)^T (\hat{P}_k^i)^{-1} \eta_k^i - \sum_{i=1}^N (\eta_{k-1}^i)^T (\hat{P}_{k-1}^i)^{-1} \eta_{k-1}^i \\ &= \sum_{i=1}^N \left(F_k^i A \eta_{k-1}^i + C_k^i u_k^i \right)^T (\hat{P}_k^i)^{-1} \left(F_k^i A \eta_{k-1}^i + C_k^i u_k^i \right) - (\eta_{k-1}^i)^T (\hat{P}_{k-1}^i)^{-1} \eta_{k-1}^i \\ &= \sum_{i=1}^N (\eta_{k-1}^i)^T \left(A^T (F_k^i)^T (\hat{P}_k^i)^{-1} F_k^i A - (\hat{P}_{k-1}^i)^{-1} \right) \eta_{k-1}^i \\ &\quad + 2 \sum_{i=1}^N (\bar{\eta}_k^i)^T (F_k^i)^T (\hat{P}_k^i)^{-1} C_k^i u_k^i + \sum_{i=1}^N (u_k^i)^T (C_k^i)^T (\hat{P}_k^i)^{-1} C_k^i u_k^i,\end{aligned} \quad (3.15)$$

where

$$u_k^i = \sum_{j \in \mathcal{N}_i} (\bar{x}_k^j - \bar{x}_k^i) = \sum_{j \in \mathcal{N}_i} (\bar{\eta}_k^j - \bar{\eta}_k^i). \quad (3.16)$$

Let us consider only the term which is not dependent on the consensus gain,

$$\Psi_k^i = -(\hat{P}_{k-1}^i)^{-1} + A^T F_k^{iT} (\hat{P}_k^i)^{-1} F_k^i A. \quad (3.17)$$

Plugging in (3.10) into (3.17) produces

$$\Psi_k^i = -(\hat{P}_{k-1}^i)^{-1} + A^T(F_k^i)^T \left(F_k^i A \hat{P}_{k-1}^i A^T (F_k^i)^T + \Pi_k^i \right)^{-1} F_k^i A, \quad (3.18)$$

where

$$\Pi_k^i = K_k^i R^i K_k^{iT} + F_k^i Q F_k^{iT}.$$

Multiplying \hat{P}_{k-1}^i on both sides of (3.18) yields

$$\hat{P}_{k-1}^i \Psi_k^i \hat{P}_{k-1}^i = \hat{P}_{k-1}^i A^T (F_k^i)^T \left(F_k^i A \hat{P}_{k-1}^i A^T (F_k^i)^T + \Pi_k^i \right)^{-1} F_k^i A \hat{P}_{k-1}^i - \hat{P}_{k-1}^i.$$

Utilizing the Woodbury matrix identity (inversion lemma) [52] and multiplying once more $(\hat{P}_{k-1}^i)^{-1}$ on both sides gives

$$\Psi_k^i = -(\hat{P}_{k-1}^i)^{-1} \left((\hat{P}_{k-1}^i)^{-1} + A^T (F_k^i)^T (\Pi_k^i)^{-1} F_k^i A \right) (\hat{P}_{k-1}^i)^{-1}. \quad (3.19)$$

Since $(\Pi_k^i)^{-1}$ and $(\hat{P}_{k-1}^i)^{-1}$ are positive definite, Ψ_k^i is negative definite. We are left to find a consensus gain such that δV_k shall always remain negative. Consider the consensus gain structure proposed by [33] of

$$C_k^i = \gamma_k \hat{P}_k^i (F_k^i)^{-T} = \gamma_k \bar{P}_k^i. \quad (3.20)$$

where the second equality stems from the well known result in Kalman filtering that $\hat{P}_k = F_k \bar{P}_k$ (see [49]). Implementing (3.20) into δV_k produces

$$\delta V_k = \sum_{i=1}^N (\eta_{k-1}^i)^T \Psi_k^i \eta_{k-1}^i + 2\gamma_k \sum_{i=1}^N (\bar{\eta}_k^i)^T u_k^i + \gamma_k^2 \sum_{i=1}^N (u_k^i)^T Y_k^i u_k^i, \quad (3.21)$$

where $Y_k^i = (F_k^i)^{-1} \hat{P}_k^i (F_k^i)^{-T}$. The second term in (3.21) can be simplified using the graph Laplacian and (3.16) as

$$2\gamma_k \sum_{i=1}^N (\bar{\eta}_k^i)^T u_k^i = -2\gamma_k \eta_{k-1}^T (L \otimes A^T A) \eta_{k-1}, \quad (3.22)$$

where η_{k-1} is the augmented agents' estimation error vector at the $k-1$ step. It is immediate that the third term in (3.21) is positive semi definite:

$$\gamma_k^2 \sum_{i=1}^N (u_k^i)^T Y_k^i u_k^i = \gamma_k^2 \eta_{k-1}^T (L \otimes A)^T Y_k (L \otimes A) \eta_{k-1}, \quad (3.23)$$

with $Y_k = \text{diag}\{Y_k^i\}_{i=1}^N$. Therefore we can write:

$$\delta V_k = -\eta_{k-1}^T \mathcal{K}_k \eta_{k-1}, \quad (3.24)$$

with

$$\mathcal{K}_k = \left(-\Psi_k + 2\gamma_k (L \otimes A^T A) - \gamma_k^2 (L \otimes A)^T Y_k (L \otimes A) \right), \quad (3.25)$$

and $\Psi_k = \text{diag}\{\Psi_k^i\}_{i=1}^N$.

We showed in (3.19) that for $\gamma_k = 0$, corresponding to the NCLKF, \mathcal{K}_k is positive definite. We now show that there must be a positive upper bound on γ_k for which \mathcal{K}_k is positive semi-definite. In this direction, we recall Sylvester's criteria [13], which states that a matrix is positive definite if and only if its leading principle minors are all positive. In this direction, let $a_1 = -[\Psi_k]_{11}$, $a_2 = [2(L \otimes A^T A)]_{11}$ and $a_3 = -[(L \otimes A)^T Y_k (L \otimes A)]_{11}$. If $a_1 + a_2\gamma_k + a_3\gamma_k^2 < 0$ (the first leading principle minor of \mathcal{K}_k is negative), then \mathcal{K}_k is not positive definite. Therefore, there must be some γ_k^* satisfying

$$0 < \gamma_k^* < \frac{-a_2 - \sqrt{a_2^2 - 4a_1a_3}}{2a_3},$$

for which the matrix \mathcal{K}_k is positive semi-definite, and for any $\gamma_k \in [0, \gamma_k^*]$, \mathcal{K}_k is positive definite and the noiseless error dynamic is asymptotically stable.

The next step in our proof is to find a method for extracting the consensus factor γ_k^* . Here, we employ the Schur complement lemma [45], which states that for matrices $W(x) = W(x)^T$, $Q(x) = Q(x)^T$ and $S(x)$ that depend affinely on x , $W(x) \succ 0$ and $Q(x) - S(x)W(x)^{-1}S(x)^T \succ 0$ if and only if

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)^T & W(x) \end{bmatrix} \succ 0.$$

Let us consider the constraint $\mathcal{K}_k(\gamma_k) \succeq 0$, and define

$$\begin{aligned} Q_k(\gamma_k) &= \Psi_k + 2\gamma_k(L \otimes A^T A) \\ W_k &= Y_k^{-1} \\ S_k(\gamma_k) &= \gamma_k(L \otimes A)^T. \end{aligned}$$

Since $[A, H^i]$ make an observable pair for all agents, the matrix F_k^i is full ranked. Additionally, we know that \hat{P}_k^i is positive-definite, therefore Y_k and its inverse are positive-definite as well. Additionally, we have $Q_k(\gamma_k) - S_k(\gamma_k)W_k^{-1}S_k(\gamma_k)^T = \mathcal{K}_k(\gamma_k)$ and thus we can conclude that $\mathcal{K}_k(\gamma_k) \succeq 0$ if and only if

$$\begin{bmatrix} \Psi_k + 2\gamma_k(L \otimes A^T A) & \gamma_k(L \otimes A)^T \\ \gamma_k(L \otimes A) & Y_k^{-1} \end{bmatrix} \succeq 0.$$

This is an LMI constraint in γ_k . We can then construct the semi-definite program

$$\begin{aligned} & \max_{\gamma_k} \gamma_k \\ & \text{s. t. } \begin{bmatrix} \Psi_k + 2\gamma_k(L \otimes A^T A) & \gamma_k(L \otimes A)^T \\ \gamma_k(L \otimes A) & Y_k^{-1} \end{bmatrix} \succeq 0, \end{aligned} \quad (3.26)$$

to obtain the largest value γ_k ensuring that \mathcal{K}_k is positive definite. This completes the proof. ■

Recall that the suggested consensus gain structure (3.20) is nothing more than the multiplication of the local matrix \bar{P}_k^i with the consensus factor γ_k . Therefore it is evident that Problem 3.2 reduces to finding the scalar γ_k which solves (3.26). This concept is manifested in Theorem 3.3. We note that in order to solve (3.26), one requires global network information such as the graph's Laplacian and the augmented matrix Y_k . In other words, the consensus factor computation is conducted in a centralized manner as illustrated in Fig. 3.4.

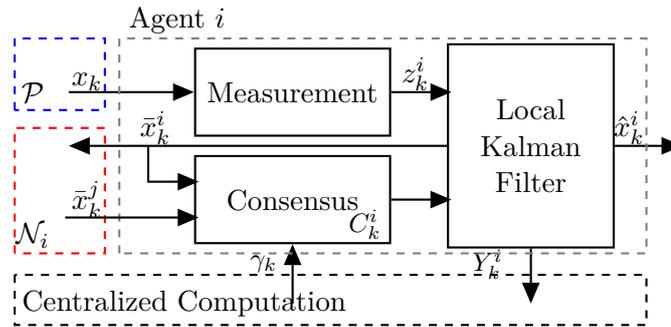


Figure 3.4: DCKE structure for the i th agent - centralized consensus gain architecture.

A comparison between the consensus gain found from (3.26) and the gain proposed in [33] is provided in Section 3.4, where superiority of our solution is demonstrated.

3.3.2 Decentralized Consensus Gain Determination

In the previous sub-section, we presented an approach for finding a consensus gain for the DCKE based on semi-definite programming. This calculation, however, must be done in a centralized manner, and the gain should be implemented for each agent in the sensor network. Note that any changes in the network structure, noise properties, or other, would require solving the SDP in (3.26) again, making this approach perhaps fragile in large-scale network systems. These points motivate an alternative method for finding a suitable consensus factor that does not require any centralized computation.

In this direction, we propose a decentralized approach for finding a suitable consensus

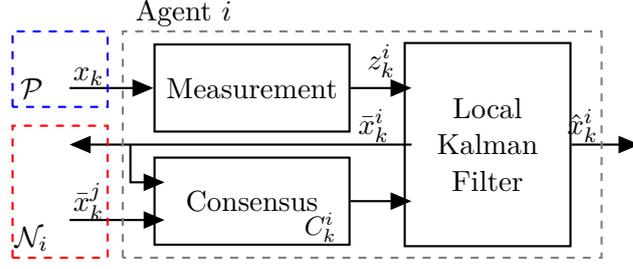


Figure 3.5: DCKE structure for the i th agent - decentralized consensus gain architecture.

gain that depends only on the local properties of the network for each agent, see Figure 3.5. In this way, we can handle time-varying graphs as well.

Consider a group of N agents, interacting over a time-varying undirected graph \mathcal{G}_k satisfying the following assumption:

Assumption 1. *The time-varying graph \mathcal{G}_k is connected at each time-instant k .*

Each sensor observes the process (3.1) with observation model (3.2). Consider now the decentralized consensus gain,

$$C_k^i = \frac{1}{|\mathcal{N}_{i,k}| + 1} F_k^i, \quad (3.27)$$

where $\mathcal{N}_{i,k}$ denotes the neighborhood of agent i at time step k . Then, the local noiseless error dynamics are

$$\begin{aligned} \eta_k^i &= F_k^i A \eta_{k-1}^i + \frac{1}{|\mathcal{N}_{i,k}| + 1} F_k^i A \sum_{j \in \mathcal{N}_{i,k}} (\eta_{k-1}^j - \eta_{k-1}^i) \\ &= F_k^i A \eta_{k-1}^i + \frac{1}{|\mathcal{N}_{i,k}| + 1} F_k^i A \sum_{j \in \mathcal{N}_{i,k}} \eta_{k-1}^j - \frac{|\mathcal{N}_{i,k}|}{|\mathcal{N}_{i,k}| + 1} \eta_{k-1}^i \\ &= \frac{1}{|\mathcal{N}_{i,k}| + 1} F_k^i A \sum_{j \in \mathcal{N}_{i,k} \cup \{i\}} \eta_{k-1}^j, \end{aligned} \quad (3.28)$$

and the augmented noiseless error dynamics are

$$\begin{aligned} \eta_k &= \text{diag}\{F_k^i A\}_{i=1}^N \left(I_{Nn} - (\mathcal{D}_k^{-1} L_k \otimes I_n) \right) \eta_{k-1} \\ &= \text{diag}\{F_k^i\}_{i=1}^N ((I_N - \mathcal{D}_k^{-1} L_k) \otimes A) \eta_{k-1}, \end{aligned} \quad (3.29)$$

with $\mathcal{D}_k = \text{diag}\{|\mathcal{N}_{i,k}| + 1\}_{i=1}^N$ and L_k denotes the graph Laplacian at time step k . It is immediate that for the non-cooperative case, i.e., when $I_N - \mathcal{D}_k^{-1} L_k = I_N$, we obtain the noiseless NCLKF error dynamics. Under the case where each sensor has the same observation of the process, we can arrive at the following result.

Proposition 3.3.1. *Assume that Assumption 1 holds and that each sensor in the*

network measures the process (3.1) using the same observation model

$$z_k^i = Hx_k + v_k^i, i = 1, \dots, N,$$

where v_k^i is the zero-mean Gaussian measurement noise with $\mathbb{E}[v_k^i(v_l^i)^T] = R\delta_{kl}$. Then the error dynamics (3.29) are asymptotically stable.

Proof.

$$\begin{aligned} \eta_k &= \text{diag}\{F_k^i A\}_{i=1}^N \left(I_{Nn} - \left(\mathcal{D}_k^{-1} L_k \otimes I_n \right) \right) \eta_{k-1} \\ &= (I_N \otimes \bar{F}_k A) \left((I_N - \left(\mathcal{D}_k^{-1} L_k \right)) \otimes I_n \right) \eta_{k-1} \\ &= \left((I_N - \left(\mathcal{D}_k^{-1} L_k \right)) \otimes \bar{F}_k A \right) \eta_{k-1}. \end{aligned}$$

Due to the properties of the Kronecker product, we have that $(I_N \otimes \bar{F}_k A)$ and $((I_N - (\mathcal{D}_k^{-1} L_k)) \otimes I_n)$ commute. This leads to the following inequality,

$$\lim_{k \rightarrow \infty} \left\| \prod_k \left((I_N - (\mathcal{D}_k^{-1} L_k)) \otimes \bar{F}_k A \right) \right\| \leq \lim_{k \rightarrow \infty} \left\| \prod_k (\bar{F}_k A) \right\| \lim_{k \rightarrow \infty} \left\| \prod_k (I_N - (\mathcal{D}_k^{-1} L_k)) \right\|.$$

From the stability of the NCLKF, it follows that $\lim_{k \rightarrow \infty} \left(\prod_k \bar{F}_k A \right) = 0$ ¹. Furthermore, it was shown in [17] that for connected graphs, the matrix $I_N - (\mathcal{D}_k^{-1} L_k)$ is ergodic at each time step k . By Wolfowitz's Theorem (see [50]), the process $\prod_k (I_N - (\mathcal{D}_k^{-1} L_k))$ is ergodic as well and there exists a vector $c \in \mathbb{R}^N$ such that

$$\lim_{k \rightarrow \infty} \prod_k (I_N - (\mathcal{D}_k^{-1} L_k)) = \mathbf{1}c^T,$$

where $\mathbf{1}$ is the matrix of all ones; Therefore,

$$\lim_{k \rightarrow \infty} \eta_k = \lim_{k \rightarrow \infty} \left(\prod_k \left((I_N - (\mathcal{D}_k^{-1} L_k)) \otimes \prod_k \bar{F}_k A \right) \right) \eta_0 = 0.$$

and the noiseless error dynamics are asymptotically stable. ■

The result of Proposition 3.3.1 may be restrictive, as we are assuming each sensor has the same measurement model with noise characteristics. On the other hand, such a model may be useful when employing a homogeneous sensor network and aiming for faster convergence of the estimate compared to using a single sensor. Currently, we do not have a proof for the general case of heterogeneous sensor measurements, however we note that in numerical simulation, over a variety of random network properties, the heterogeneous case gives promising results. We explore this in the next section.

The above proposition provides an extremely simple method to find a consensus factor

¹Here we use an abuse of conventional notation and define $\prod_{k=1}^n M_k = M_n M_{n-1} \dots M_2 M_1$.

that works. In contrast, [39] proposed the following decentralized consensus gain,

$$C_k^i = \frac{\epsilon}{1 + |\hat{P}_k^i|_F} \hat{P}_k^i, \quad (3.30)$$

where ϵ is some predetermined constant. The design constant ϵ can be pre-calibrated, however no mid-run modification techniques were provided for this constant in case of, for example, a change in the graph structure.

Remark 1. *It should be noted that the consensus gain structure in the centralized scheme of Theorem 3.3 is not the same as the one proposed in (3.27). The centralized consensus gain was chosen to ensure explicitly that the Lyapunov function (3.14) decreases along the system trajectories. On the other hand, the decentralized gain was chosen to simplify the structure of the error dynamics. Although not having the same structure, we would expect that the centralized consensus gain found using (3.26) would out perform the proposed decentralized consensus gain. This is due to the fact that the centralized estimator employs global network properties to compute the consensus gain, whereas, in the decentralized scheme, only local network properties are employed. For the numerical example presented in Section 3.4 we note that this is not the case, and in fact the decentralized scheme shows better results.*

3.4 Simulation Results

The following numerical example was taken from [34] with minor modifications. Consider a robot performing a noisy “snail” trajectory with the following dynamics,

$$x_{k+1} = \underbrace{\begin{bmatrix} 0.9996 & -0.0283 \\ 0.0283 & 0.9996 \end{bmatrix}}_A x_k + \underbrace{0.375 \cdot I_2}_B w_k. \quad (3.31)$$

The robot’s initial state is set to be $x_0 = [15, -10]^T$, the initial covariance matrix for each agent is set to be $P_0^i = 10I$, and the agents’ initial estimates are normally distributed about the initial state. Additionally, the process noise covariance is $Q = I_2$. A network of 20 sensors are randomly positioned in some field of interest (see Fig. 3.6) where a communication link between 2 sensors exists only if their distance is below some threshold (< 40 meters). Furthermore, we consider two sensing models: 1) the homogeneous sensing model where each agent measure the robot with the same observation model such that $R^i = R = 9$ and $H^i = H = [0.5, 0.5]$, and 2) the heterogeneous model where each agent with an even number measures the robot’s y -axis position

while the agents with an odd number measure its x -axis position such that:

$$H^i = \begin{cases} [1, 0] & i \in \{1, 3, \dots, 19\} \\ [0, 1] & i \in \{2, 4, \dots, 20\} \end{cases}. \quad (3.32)$$

The measurement noise covariance for the i th agent is $R^i = \sqrt{i}$.

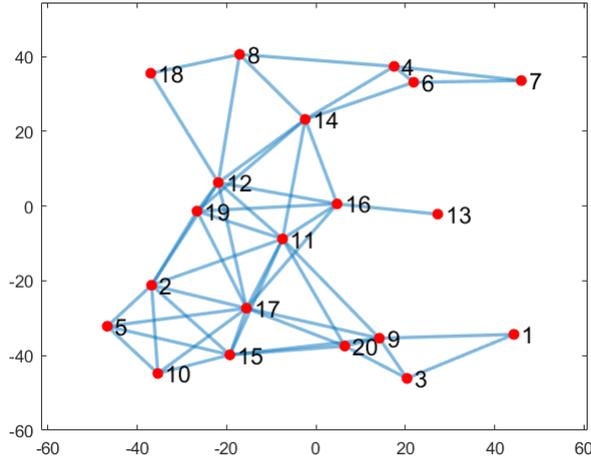


Figure 3.6: A sensor network of 20 agents randomly positioned.

We provide a comparison between 7 state estimators:

NCLKF: the non-cooperative local Kalman filter with null consensus gain;

SOCKF1: the sub-optimal consensus Kalman filter with consensus factor (3.12);

SOCKF2: the sub-optimal consensus Kalman filter with consensus factor (3.26) (computed utilizing CVX toolbox [14]);

DSOCKF1: the decentralized sub-optimal consensus Kalman filter with $\epsilon = 0.1$ and consensus gain (3.30);

DSOCKF2: the decentralized sub-optimal consensus Kalman filter with consensus gain (3.27);

OCKF: the optimal consensus Kalman filter (3.6) with the optimal consensus gain (3.9).

CENKF: centralized Kalman filter where all agents' measurements are processed in a single Kalman filter.

The compared performance measures are twofold: the agents state estimation standard

deviation (Fig. 3.7) calculated as

$$\sigma^x = \frac{1}{MC} \sum_{j=1}^{MC} \sqrt{\frac{1}{N-1} \sum_{i=1}^N \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \hat{x}^{i,j} - \frac{1}{N} \begin{bmatrix} 1 & 0 \end{bmatrix} \sum_{i=1}^N \hat{x}^{i,j} \right)^2}$$

$$\sigma^y = \frac{1}{MC} \sum_{j=1}^{MC} \sqrt{\frac{1}{N-1} \sum_{i=1}^N \left(\begin{bmatrix} 0 & 1 \end{bmatrix} \hat{x}^{i,j} - \frac{1}{N} \begin{bmatrix} 0 & 1 \end{bmatrix} \sum_{i=1}^N \hat{x}^{i,j} \right)^2},$$

where MC denotes the number of Monte-Carlo runs and $\hat{x}^{i,j}$ is the i th agent state estimation for the j th run. The true averaged root mean squared error (Fig. 3.8) calculated as

$$\text{RMSE} = \frac{1}{MC} \sum_{j=1}^{MC} \sqrt{\sum_{i=1}^N (\mathbb{E}[(\eta^{i,j})^T \eta^{i,j}])},$$

where $\eta^{i,j} = \hat{x}^{i,j} - x$.

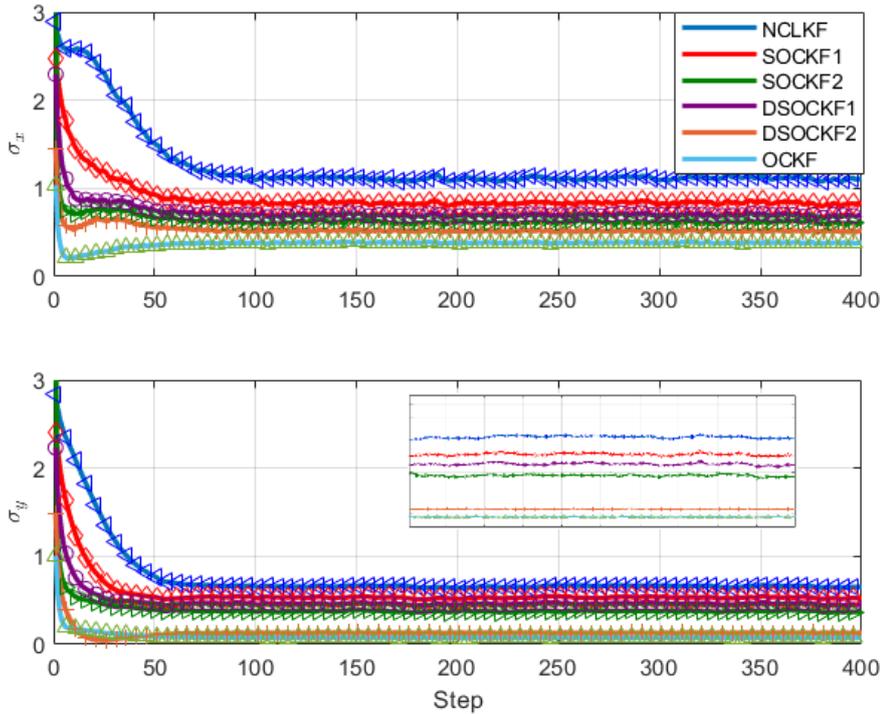


Figure 3.7: Standard deviation of the agents' state estimation for both axes, comparison between 6 distributed state estimators over 100 Monte-Carlo runs for a homogeneous sensing model.

We begin our discussion with the homogeneous sensing scheme, in Fig. 3.7 one can observe the agents rate of convergence and stability of the agents' estimation error for all 6 distributed state estimators (all except CENKF). As shown, the OCKF and the DSOCKF2 converges with the fastest rate among all estimators. Additionally shown the relative proximity of the SOCKF1 to the NCLKF estimator performance. This results

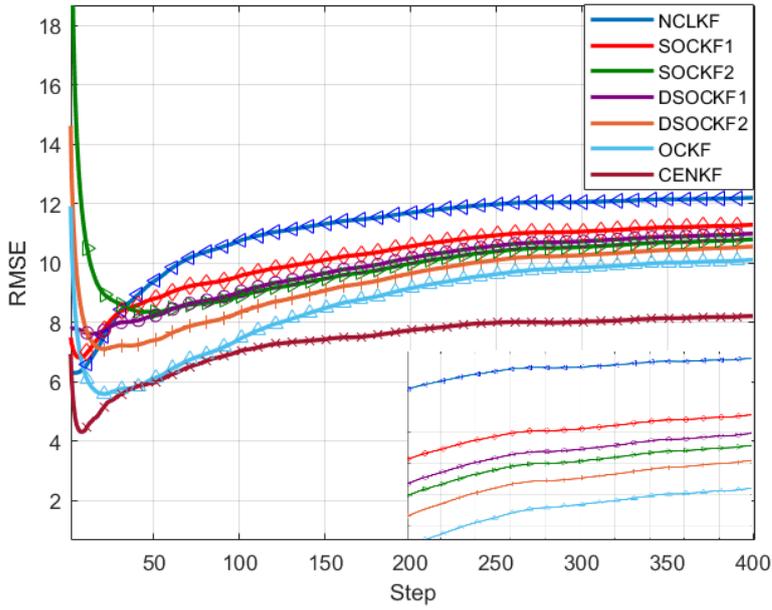


Figure 3.8: Root mean squared error, comparison between 7 state estimators over 100 Monte-Carlo runs for a homogeneous sensing model.

due to the extremely small consensus factor gain used by SOCKF1 which effectively ignores the effect of the consensus component, thus turning SOCKF1 into a NCLKF estimator.

Fig. 3.8 demonstrates the superiority of SOCKF2 and DSOCKF2 over other solutions by presenting lower root mean squared error (excluding the optimal solutions). Their superiority over the NCLKF is expected as there is more information for the agents to process. Additionally shown that for this specific graph topology, the SOCKF2 shows superiority over the DSOCKF1, potentially due to a poor selection of the gain ϵ . The CENKF represents the best obtainable solution while the OCKF represents a lower bound on our proposed sub-optimal and distributed consensus based strategies. In these simulations we see the gap between our solutions and OCKF/CENKF is not significant.

Although the DSOCKF2 was proven for the homogeneous model, we show that in fact it provides satisfying results for the heterogeneous model as well. We note that, in this scheme, although (A, H^i) make an observable pair for each individual sensor, the observability is relatively weak for the non-measured axis, i.e., while the robot is in transition between quadrants one would expect a relatively large estimation error since the position in one axis hardly vary while the position in the other can vary significantly.

In Fig. 3.9 we see the standard deviation of the agents' estimation for the heterogeneous model. As shown, the SOCKF2 converges with the fastest rate among the centralized

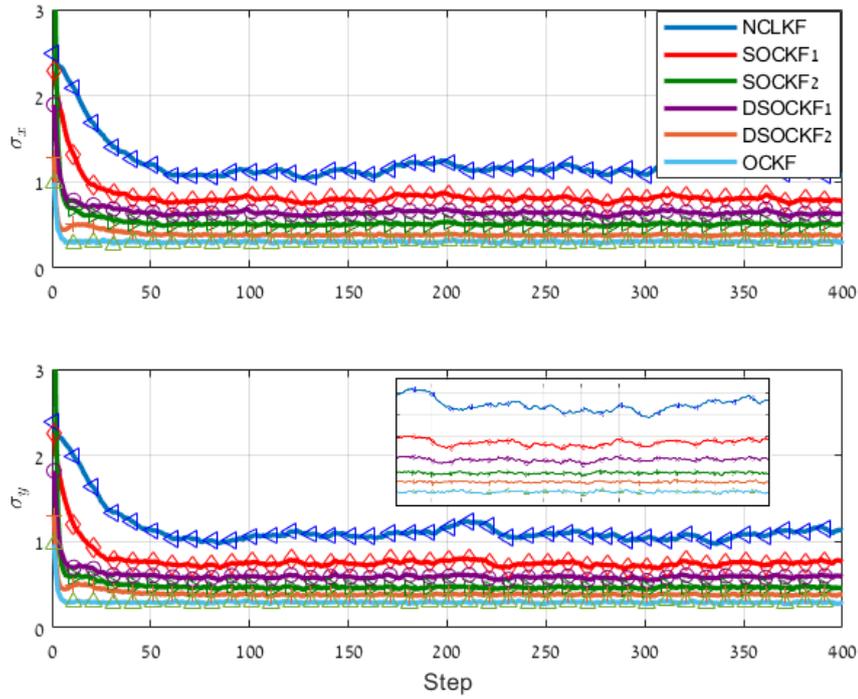


Figure 3.9: Standard deviation of the agents' state estimation for both axes, comparison between 6 state estimators over 100 Monte-Carlo runs for a heterogeneous sensing model.

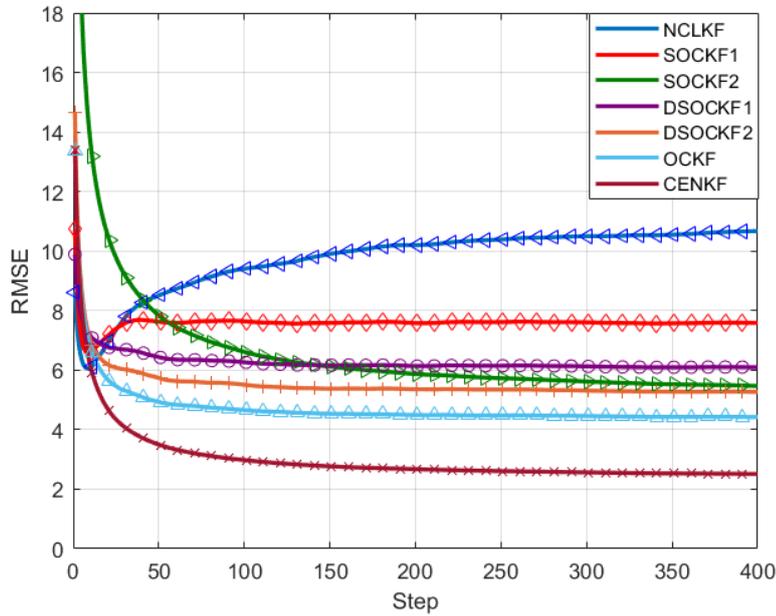


Figure 3.10: Root mean squared error, comparison between 7 state estimators over 100 Monte-Carlo runs for a heterogeneous sensing model.

filters while maintaining a relatively constant state estimation standard deviation (even through quadrants transition). Additionally, one can observe that, in this scheme

as well, the SOCKF1 does not show drastic improvement over the NCLKF. In the decentralized schemes we observe superiority of the DSOCKF2 over DSOCKF1 here as well, and as expected, the OCKF outperformed all other estimators.

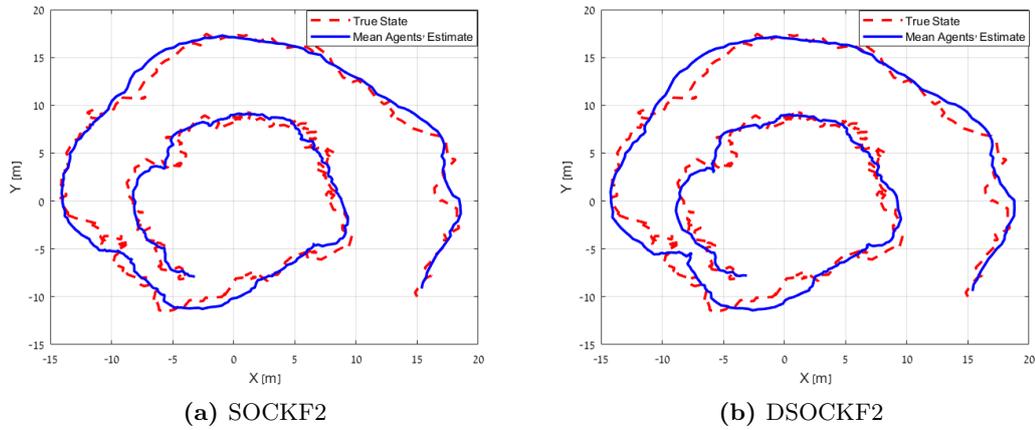


Figure 3.11: Trajectory of the true state and the agents' mean estimate utilizing SOCKF2 (a) and DSOCKF2 (b) for a heterogeneous sensing model.

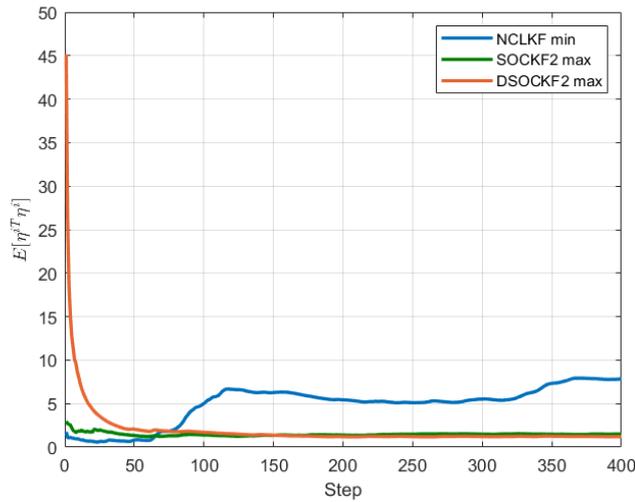


Figure 3.12: Local MSE, comparison between agents with minimum NCLKF MSE, maximum SOCKF2 MSE and maximum DSOCKF2 MSE for a single run with a heterogeneous sensing model.

Fig. 3.10 further demonstrates the superiority of SOCKF2 and DSOCK2 over the others by presenting lower root mean squared error. Once more it is shown that for this specific graph topology, the SOCKF2 shows superiority over the DSOCKF1. Additionally shown is that, here as well, the gap between our suggested solution and the OCKF/CENKF is not large. What is perhaps most astonishing is that these simulations indicate the decentralized consensus gain selection for DSOCKF2 outperforms the centralized consensus factor in SOCKF2. This result can be explained emphasizing the following two points regarding the optimal gain (3.9). Firstly, (3.9) is a decentralised

solution. Secondly, both (3.9) and (3.27) are of the form $C_k^i = F_k^i \Theta_k^i$, where Θ_k^i is computed differently for each gain. In (3.9), $\Theta_k^i = (\Gamma_k^i)^T \left(\Gamma_k^i (H^i)^T (\Omega_k^i)^T (\Gamma_k^i)^T - D_k^i \right)^{-1}$ whereas in (3.27), $\Theta_k^i = \frac{1}{|\mathcal{N}_{i,k}|+1} I_n$. Fig. 3.11 illustrates the true and mean estimated trajectory of the robot using SOCKF2 (3.11a), and DSOCKF2 (3.11b) for the heterogeneous sensing model. As shown, the proposed filter provides good tracking results.

Fig. 3.12 provides zoom-in demonstration for the effectiveness of our proposed solutions with respect to the local estimator performance. This is done by comparing between agents which obtained the maximum local MSE for estimators SOCKF2 and DSOCKF2 with the agent which obtained the minimum local MSE for NCLKF. As shown, local performance improved drastically with our proposed solutions.

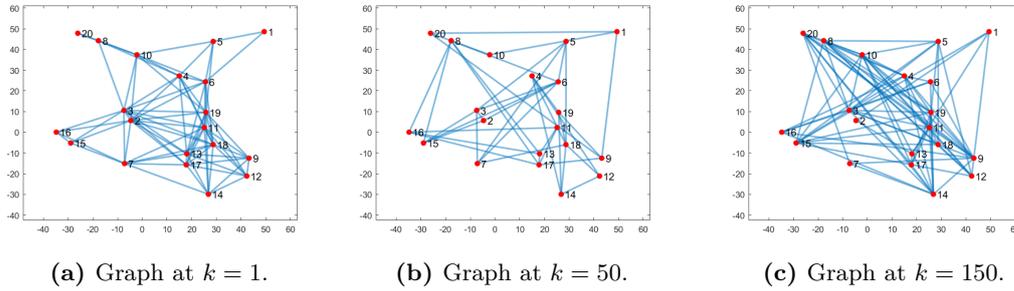


Figure 3.13: Communication graph at time steps (a) 1-49, (b) 50-149, and (c) 150-300.

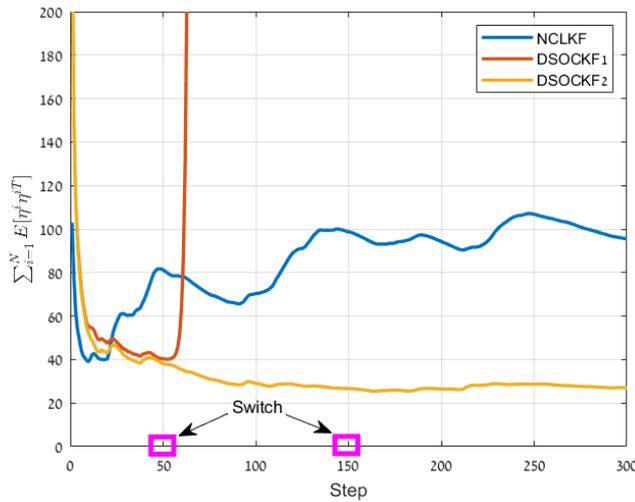


Figure 3.14: Sum of all agents mean squared error with two graph switches (at step 50 and at step 150), comparison between 3 state estimators for a single run with the heterogeneous sensing model.

To conclude, we compare the robustness of the proposed decentralized consensus Kalman filter to the filter proposed in [39]. To do so, we simulate a communication topology switch at two time instances, at $k = 50$ and $k = 150$ (see Fig. 3.13). The sum of

all agents MSE are presented in Fig. 3.14, where we compare between 3 estimators: NCLKF, DSOCKF1, and DSOCKF2. As shown, the DSOCKF1 becomes unstable after the first switch, while the DSOCKF2 remains stable for the entire duration.

To summarise this chapter, we have demonstrated superiority of our proposed consensus gains compared to existing solutions and the non-cooperative Kalman estimator, in both the homogeneous and the heterogeneous schemes. We did so for two consensus gain computation methods: centralized and the decentralised. We further demonstrated robustness to time varying communication topology for the decentralised method.

Chapter 4

Event-Triggered Consensus Kalman Filter

In the previous chapter we introduced the consensus Kalman filter and showed superiority in performance over the NCLKF. In this chapter we shall expand our discussion to include energy consumption in the form of communication load reduction for each sensor in the network. To do so we shall implement an event-triggering mechanism in the consensus Kalman filter while preserving aspects such as stability and performance compared to the NCLKF.

4.1 Problem Setup

We consider a network comprising N interacting sensor agents where the interaction topology can be described by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ indicates which agents can exchange information with each other. Each agent observes a linear discrete-time stochastic process described by the dynamics

$$\mathcal{P} : x_{k+1} = Ax_k + Bw_k, \quad (4.1)$$

where $x_k \in \mathbb{R}^n$ is the state vector and w_k is an additive white Gaussian noise such that $\mathbb{E} [w_k w_l^T] = Q\delta_{kl}$, where δ_{kl} is the Dirac Delta function.

Each agent is capable of measuring the process state using the observation model

$$z_k^i = H^i x_k + v_k^i, \quad (4.2)$$

where $z_k^i \in \mathbb{R}^{m^i}$ is the measurement obtained by agent i , $H^i \in \mathbb{R}^{m^i \times n}$ is the observation matrix, and $v_k^i \in \mathbb{R}^{m^i}$ is a measurement noise assumed to also be additive white

Gaussian noise with $\mathbb{E} \left[v_k^i (v_k^i)^T \right] = R^i \delta_{kl}$. Additionally we assume that $R^i \in \mathbb{R}^{m^i \times m^i}$ is invertible and that (A, H^i) make an observable pair for every agent such that the noiseless NCLKF is asymptotically stable.

The distributed *event-triggered consensus Kalman estimator* (DETCKE) was first proposed by [48],

$$\hat{x}_k^i = \bar{x}_k^i + K_k^i \left(z_k^i - H^i \bar{x}_k^i \right) + C_k^i \sum_{j \in \mathcal{N}_i} \left(\tilde{x}_k^{ji} - \tilde{x}_k^{ii} \right), \quad (4.3)$$

where K_k^i and C_k^i are the Kalman and consensus gains of the i th agent, respectively, and \hat{x}^i and \bar{x}^i are the *posteriori* and *a priori* state estimate of the i th agent, respectively. Additionally, \tilde{x}^{ji} denote the j th agent state propagation, used in the i th agent estimator, according to the following event triggering rule,

$$\tilde{x}_k^{ji} = \begin{cases} \bar{x}_k^j, & f_k^j \left(\{ \tilde{x}_k^{sj} \}_{s \in \mathcal{N}_j \cup \{j\}}, \bar{x}_k^j \right) > 0 \\ A \tilde{x}_{k-1}^{ji}, & f_k^j \left(\{ \tilde{x}_k^{sj} \}_{s \in \mathcal{N}_j \cup \{j\}}, \bar{x}_k^j \right) \leq 0 \end{cases} \quad \forall j \in \mathcal{N}_i \cup \{i\}, \quad (4.4)$$

where $f_k^j \left(\{ \tilde{x}_k^{sj} \}_{s \in \mathcal{N}_j \cup \{j\}}, \bar{x}_k^j \right)$ is the event triggering function comprising local network properties and locally computed estimates, evaluated by the j th agent in each step. Formally, if the ETC $f_k^j \left(\{ \tilde{x}_k^{sj} \}_{s \in \mathcal{N}_j \cup \{j\}}, \bar{x}_k^j \right) \leq 0$ is satisfied, then the i th agent will continue propagating the most recent state prediction obtained from the j th agent (assuming they are neighbors). If the ETC is violated then the j th agent will broadcast its current state prediction. The function $f_k^j \left(\{ \tilde{x}_k^{sj} \}_{s \in \mathcal{N}_j \cup \{j\}}, \bar{x}_k^j \right)$ holds a key role in the filter update equations. If not well formulated, this function could lead to the absence of events which may cause the local estimation errors to diverge. On the other hand, it may generate an event at each time instance such that the event-triggered consensus Kalman estimator, for all practical purposes, serves as the consensus Kalman estimator (3.3).

The event-triggered consensus Kalman estimator (4.3) is composed of a classic Kalman estimator term, a consensus term based on neighbors last transmitted estimates and an event-triggered mechanism. As Fig. 4.1 suggests, the ETM is responsible for the propagation of the state from the most recent event, \tilde{x}_k^{ii} . It should be noted that, although not captured in this figure, the ETM shall manage the state prediction broadcasting when the ETC is violated .

4.2 Event-Triggered Consensus Kalman Estimator

In this section we explore an event-triggered mechanism to reduce transmission overloads while ensuring stability of the consensus Kalman filter. Essentially we seek to

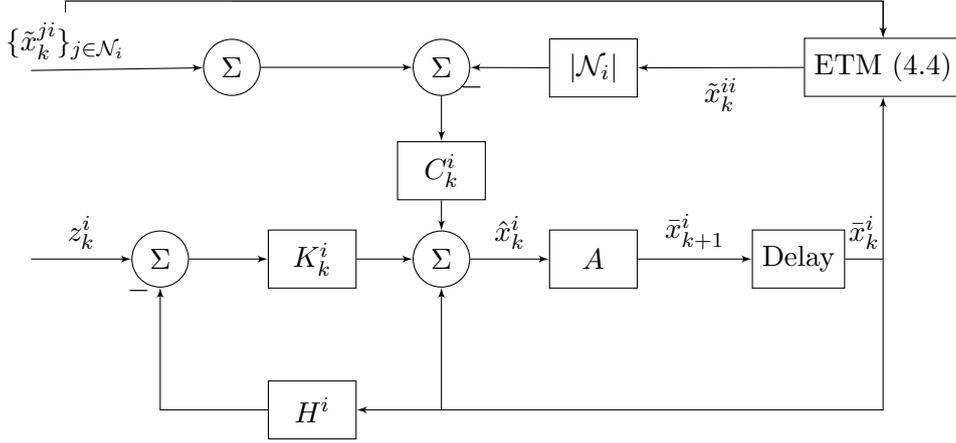


Figure 4.1: DETCKE of the i^{th} agent.

solve the following problem:

Problem 4.1 (Sub-optimal event-triggered consensus Kalman filter). *Consider N agents interacting over a connected graph \mathcal{G} where each observes the process (4.1) with observation model (4.2). Additionally, each agent utilizes an estimator of type (4.3) and an event triggering rule of type (4.4). Using a one-hop based information exchange, find a set of local gains, K_k^i and C_k^i , and an ETF, $f_k^i(\cdot)$, that will ensure the stability of the estimation error and reduce communication bandwidth.*

Once more, our discussion is divided into two approaches: centralized and decentralized methods for designing the consensus gain C_k^i .

4.2.1 Event-Triggered Condition for a Centralized Consensus Gain

We now consider the consensus gain (3.20) with the event-triggered condition for the i th agent, as proposed in [26],

$$f_k^i(\{\tilde{x}_k^{ji}\}_{j \in \mathcal{N}_i \cup \{i\}}, \bar{x}_k^i) = (\bar{x}_k^i - \tilde{x}_k^{ii})^T \sum_{j \in \mathcal{N}_i} (\tilde{x}_k^{ji} - \tilde{x}_k^{ii}) < 0. \quad (4.5)$$

Next we will propose a range of values for the consensus factor, with which we solve Problem 4.1 and ensure the stability of the error dynamics. This is presented in the following theorem.

Theorem 4.2 (DETCKE Stability). *Consider a group of N agents interacting over a connected graph \mathcal{G} where each observes the process (4.1) with observation model (4.2). The noiseless estimation error for the consensus Kalman filter (3.10) with the DETCKE (4.3), the event-triggered mechanism (4.4) with the event-triggered condition (4.5), the*

choice of consensus gain $C_k^i = \gamma_k P_k^i (F_k^i)^{-T}$ and any consensus factor satisfying

$$0 \leq \gamma_k \leq \frac{2}{\lambda_{max} \left(\text{diag} \left\{ (F_k^i)^{-1} P_k^i (F_k^i)^{-T} \right\}_{i=1}^N \right) \lambda_{max}(L)},$$

is asymptotically stable.

Proof. To prove the stated theorem, first we choose a quadratic Lyapunov function and show that for the suggested consensus factor and event-triggered condition, the Lyapunov function is monotonically decreasing.

Let $\eta_k = \hat{x}_k - x_k$, $\bar{\eta}_k = \bar{x}_k - x_k$ and $\tilde{\eta}_k = \tilde{x}_k - x_k$ be the estimation, prediction, and propagation errors, respectively. The noiseless error dynamics are

$$\begin{aligned} \eta_k^i &= \underbrace{(I - K_k^i H^i)}_{F_k^i} \bar{\eta}_k^i + C_k^i \sum_{j \in \mathcal{N}_i} (\tilde{\eta}_k^{ji} - \tilde{\eta}_k^{ii}) \\ \bar{\eta}_{k+1}^i &= A \eta_k^i. \end{aligned}$$

For convenience, we define the following notation $\tilde{u}_k^i = \sum_{j \in \mathcal{N}_i} (\tilde{\eta}_k^{ji} - \tilde{\eta}_k^{ii})$. Consider now the following Lyapunov function,

$$V_k = \sum_{i=1}^N (\eta_k^i)^T (\hat{P}_k^i)^{-1} \eta_k^i. \quad (4.6)$$

The Lyapunov difference along the system trajectories is

$$\begin{aligned} \delta V_k &= V_k - V_{k-1} = \sum_{i=1}^N (\eta_k^i)^T (\hat{P}_k^i)^{-1} \eta_k^i - \sum_{i=1}^N (\eta_{k-1}^i)^T (\hat{P}_{k-1}^i)^{-1} \eta_{k-1}^i \\ &= \sum_{i=1}^N \left(F_k^i A \eta_{k-1}^i + C_k^i \tilde{u}_k^i \right)^T (\hat{P}_k^i)^{-1} \left(F_k^i A \eta_{k-1}^i + C_k^i \tilde{u}_k^i \right) - (\eta_{k-1}^i)^T (\hat{P}_{k-1}^i)^{-1} \eta_{k-1}^i \\ &= \sum_{i=1}^N (\eta_{k-1}^i)^T \left(A^T (F_k^i)^T (\hat{P}_k^i)^{-1} F_k^i A - (\hat{P}_{k-1}^i)^{-1} \right) \eta_{k-1}^i \\ &\quad + 2 \sum_{i=1}^N (\bar{\eta}_k^i)^T (F_k^i)^T (\hat{P}_k^i)^{-1} C_k^i \tilde{u}_k^i + \sum_{i=1}^N (\tilde{u}_k^i)^T (C_k^i)^T (\hat{P}_k^i)^{-1} C_k^i \tilde{u}_k^i, \end{aligned}$$

It was shown in the proof of Theorem 3.3 that the matrix $\Psi_k^i = A^T (F_k^i)^T (\hat{P}_k^i)^{-1} F_k^i A - (\hat{P}_{k-1}^i)^{-1}$ satisfies

$$\Psi_k^i < 0, \quad (4.7)$$

i.e., it is negative definite. Recall the consensus gain structure from (3.20),

$$C_k^i = \gamma_k \hat{P}_k^i (F_k^i)^{-T} = \gamma_k \bar{P}_k^i. \quad (4.8)$$

Inserting (4.8) into δV_k produces

$$\delta V_k = \sum_{i=1}^N (\eta_{k-1}^i)^T \Psi_k^i \eta_{k-1}^i + 2\gamma_k \sum_{i=1}^N (\bar{\eta}_k^i)^T \tilde{u}_k^i + \gamma_k^2 \sum_{i=1}^N (\tilde{u}_k^i)^T Y_k^i \tilde{u}_k^i, \quad (4.9)$$

where $Y_k^i = (F_k^i)^{-1} \hat{P}_k^i (F_k^i)^{-T}$. We now consider the event trigger condition for the i th agent,

$$f_k^i \left(\{\tilde{x}_k^{ji}\}_{j \in \mathcal{N}_i \cup \{i\}}, \bar{x}_k^i \right) = \left(\bar{x}_k^i - \tilde{x}_k^{ii} \right)^T \tilde{u}_k^i = \left(\bar{\eta}_k^i - \tilde{\eta}_k^{ii} \right)^T \tilde{u}_k^i \leq 0. \quad (4.10)$$

If ETC (4.10) is violated, then according to ETM (4.4) we obtain that $\bar{x}_k^i = \tilde{x}_k^{ii}$ or equivalently $\bar{\eta}_k^i = \tilde{\eta}_k^{ii}$ such that:

$$2\gamma_k (\bar{\eta}_k^i)^T \tilde{u}_k^i = 2\gamma_k (\tilde{\eta}_k^{ii})^T \tilde{u}_k^i.$$

If ETC (4.10) is satisfied, then the following inequality holds:

$$2\gamma_k (\bar{\eta}_k^i)^T \tilde{u}_k^i \leq 2\gamma_k (\tilde{\eta}_k^{ii})^T \tilde{u}_k^i, \quad (4.11)$$

thus the second term in (4.9) satisfies

$$2\gamma_k \sum_{i=1}^N (\bar{\eta}_k^i)^T \tilde{u}_k^i \leq 2\gamma_k \sum_{i=1}^N (\tilde{\eta}_k^{ii})^T \tilde{u}_k^i, \quad (4.12)$$

for each time step k . Additionally, we know that \tilde{x}^{ji} is the same for all $i \in \mathcal{N}_j$, i.e.,

$$\tilde{x}^{ji} = \tilde{x}^{js} \forall \{i, s\} \in \mathcal{N}_j,$$

such that the following equality holds,

$$\tilde{u}_k = (L \otimes I_n) \tilde{\eta}, \quad (4.13)$$

with $\tilde{u}_k = \left[(\tilde{u}_k^1)^T, (\tilde{u}_k^2)^T, \dots, (\tilde{u}_k^N)^T \right]^T$. Fusing (4.7), (4.12), and (4.13) into (4.9) yields

$$\begin{aligned} \delta V_k &= \sum_{i=1}^N (\eta_{k-1}^i)^T \Psi_k^i \eta_{k-1}^i + 2\gamma_k \sum_{i=1}^N (\bar{\eta}_k^i)^T \tilde{u}_k^i + \gamma_k^2 \sum_{i=1}^N (\tilde{u}_k^i)^T Y_k^i \tilde{u}_k^i \\ &< 2\gamma_k \sum_{i=1}^N (\tilde{\eta}_k^{ii})^T \tilde{u}_k^i + \gamma_k^2 \lambda_{\max}(Y_k) \sum_{i=1}^N (\tilde{u}_k^i)^T \tilde{u}_k^i \\ &\leq -2\gamma_k \tilde{\eta}_k^T L \tilde{\eta}_k + \gamma_k^2 \lambda_{\max}(Y_k) \tilde{\eta}_k^T L L \tilde{\eta}_k \\ &\leq -(2\gamma_k - \gamma_k^2 \lambda_{\max}(Y_k) \lambda_{\max}(L)) \tilde{\eta}_k^T L \tilde{\eta}_k \leq 0, \end{aligned}$$

where $Y_k = \text{diag}\{Y_k^i\}_{i=1}^N$. Hence, for the consensus factor

$$0 \leq \gamma_k \leq \frac{2}{\lambda_{\max}(Y_k)\lambda_{\max}(L)}, \quad (4.14)$$

the Lyapunov difference (4.9) is strictly decreasing. ■

It should be noted that by choosing $\gamma_k = 0$, we obtain the NCLKF for which stability is ensured. We have shown that by selecting a consensus factor which satisfies (4.14), consensus gain structure (3.20), and ETC (4.10), we are able to guaranty the stability of the noiseless estimation error.

Remark 2. *The consensus gain proposed in [26] has the form*

$$C_k^i = \frac{2F_k^i\Gamma_k^{i-1}}{\lambda_{\max}(\Gamma_k^{-1})\lambda_{\max}(L)}, \quad (4.15)$$

where $\Gamma_k^i = (F_k^i)^T A^T (\bar{P}_k^i)^{-1} A F_k^i$ and $\Gamma_k = \text{diag}\{\Gamma_k^i\}_{i=1}^N$. The computation of (4.15) requires the assumption that the matrix A is non-singular. However, even if it is non-singular but ill-conditioned, it may still lead to numerical challenges. This is in contrast to the gain proposed in this work that does not require inversion of the process dynamics. The performance degradation of (4.15) with respect to the consensus gain stated in Theorem 4.2 will be presented in Section 4.4.

In order to solve (4.14), one requires global network information such as the graph's Laplacian matrix and the augmented matrix Y_k . Therefore, the consensus factor computation must be conducted in a centralized manner as illustrated in Fig. 4.2.

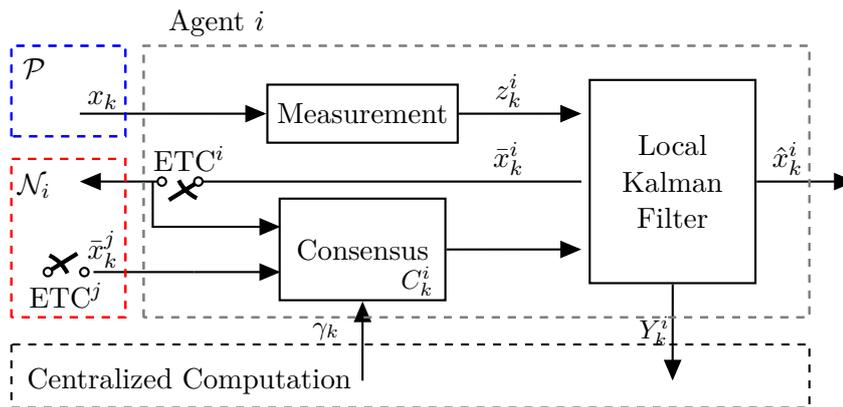


Figure 4.2: DETCKE structure for the i th agent - centralized consensus gain architecture.

We now seek to construct a solution which does not requires global network information.

4.2.2 Event-Triggered Condition for a Decentralized Consensus Gain

In the previous subsection we proposed a centralized solution for the determination of the consensus factor γ_k that required global network information, i.e., the largest eigenvalue of the Laplacian matrix L and the augmented matrix Y_k . In Chapter 3, we showed that this choice of gain can be problematic for time-varying communication networks. Additionally, for large-scale networks, the centralized architecture may also require heavy computational tools. In this subsection, we propose a decentralized event-trigger scheme in which the event-triggering function and the consensus factor are formulated such that only local network properties are required (see Fig. 4.3).

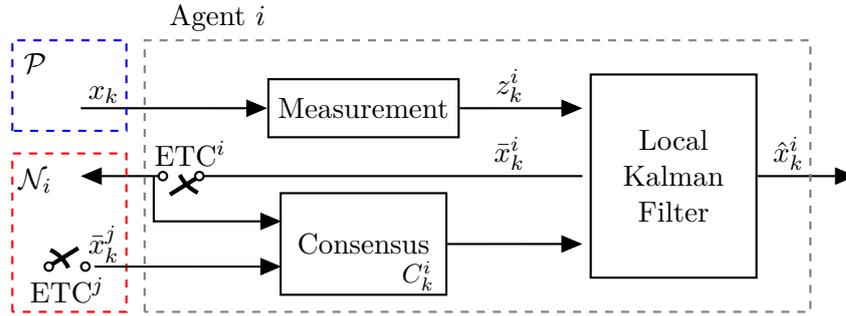


Figure 4.3: DETCKE structure for the i th agent - decentralized consensus gain architecture.

Consider a group of N agents, interacting over a time-varying and undirected communication graph, denoted by \mathcal{G}_k . Different from the setup in Section 3.3.2, here we do not require that \mathcal{G}_k is connected at each step. Each sensor observes the process (4.1) with observation model (4.2) and the consensus Kalman estimator:

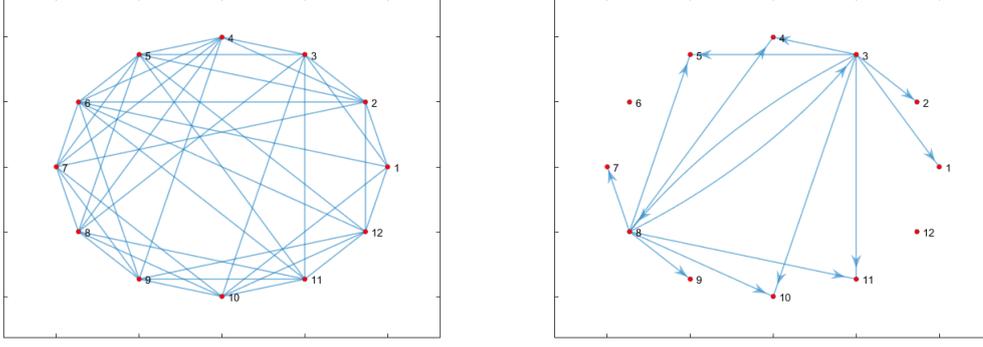
$$\hat{x}_k^i = \bar{x}_k^i + K_k^i \left(z_k^i - H^i \bar{x}_k^i \right) + C_k^i \sum_{j \in \mathcal{N}_{i,k}} \beta_k^j \left(\bar{x}_k^j - \bar{x}_k^i \right), \quad (4.16)$$

where $\mathcal{N}_{i,k}$ is neighborhood of the i th agent with respect to the graph \mathcal{G}_k and $\beta_k^j = 1$ if the j th agent broadcasts its information and $\beta_k^j = 0$, otherwise. In (4.16), we use the broadcasting neighbors *a priori* state prediction in the consensus term instead of the state propagation \tilde{x}_k^{jj} of non-broadcasting agents as formulated in (4.3). In this direction, we introduce the following ETM:

$$\beta_k^j = \begin{cases} 1, & f_k^j \left(\{ \tilde{x}_k^{sj} \}_{s \in \mathcal{N}_j \cup \{j\}}, \bar{x}_k^j \right) > 0 \\ 0, & f_k^j \left(\{ \tilde{x}_k^{sj} \}_{s \in \mathcal{N}_j \cup \{j\}}, \bar{x}_k^j \right) \leq 0 \end{cases} \quad \forall j \in \mathcal{N}_i \cup \{i\}. \quad (4.17)$$

Additionally, we consider the SoD event-triggering condition for the j th agent:

$$f_k^j \left(\{ \tilde{x}_k^{sj} \}_{s \in \mathcal{N}_j \cup \{j\}}, \bar{x}_k^j \right) = \left(\tilde{x}_k^{jj} - \bar{x}_k^j \right)^T \left(\tilde{x}_k^{jj} - \bar{x}_k^j \right) - \delta \leq 0, \quad (4.18)$$



(a) Undirected graph - \mathcal{G}_k .

(b) Directed graph - \mathcal{G}_k^* .

Figure 4.4: Available v.s instantaneous communication topology where agents broadcast information based on an event triggering mechanism.

where \tilde{x}^{jj} is formulated as in (4.4) and $\delta \in \mathbb{R}$ is some constant threshold so that an agent will broadcast (trigger an event) if the error norm between the state prediction and the state propagation exceeds some value.

To simplify future discussion, we introduce a new notation set to describe the instantaneous broadcast channels of an event-triggering scheme. Let L_k be the Laplacian matrix for the sensing network \mathcal{G}_k describing the available communication channels between each agent at step k . Let $\mathcal{G}_k^* = (\mathcal{V}, \mathcal{E}_k^*)$ denote the transmission graph of agents for which an event was triggered such that $\mathcal{E}_k^* = \{(j, i) \mid \beta_k^j = 1, j \in \mathcal{N}_{i,k}\}$. For example, Fig. 4.4b illustrates the instantaneous communication topology in which only the 3rd and the 8th agents broadcast, while Fig. 4.4a depicts the available communication topology, \mathcal{G}_k . Additionally, let L_k^* be the Laplacian matrix for the directed sensing network \mathcal{G}_k^* . Finally let $\mathcal{N}_{i,k}^*$ be the local neighborhood of the i th agent with respect to \mathcal{G}_k^* such that $\mathcal{N}_{i,k}^* = \{j \mid \beta_k^j = 1, i \in \mathcal{N}_{j,k}\}$. Thus, (4.16) can be reformulated as such:

$$\hat{x}_k^i = \bar{x}_k^i + K_k^i \left(z_k^i - H^i \bar{x}_k^i \right) + C_k^i \sum_{j \in \mathcal{N}_{i,k}^*} \left(\bar{x}_k^j - \bar{x}_k^i \right). \quad (4.19)$$

We now consider the following consensus gain

$$C_k^i = \begin{cases} \frac{1}{|\mathcal{N}_{i,k}^*| + 1} F_k^i, & |\mathcal{N}_{i,k}^*| > 0 \\ 0, & |\mathcal{N}_{i,k}^*| = 0 \end{cases}, \quad (4.20)$$

Note that for the case where an agent has no broadcasting neighbors, i.e., $|\mathcal{N}_{i,k}^*| = 0$, the consensus term will be zero and it will run the NCLKF. In this case it is straightforward

that the noiseless error dynamics are

$$\begin{aligned}\eta_k^i &= F_k^i \bar{\eta}_k^i \\ \bar{\eta}_{k+1}^i &= A \eta_k^i.\end{aligned}\tag{4.21}$$

For the non-empty neighborhood case, the local noiseless error dynamics are

$$\begin{aligned}\eta_k^i &= F_k^i \bar{\eta}_k^i + \frac{1}{|\mathcal{N}_{i,k}^*| + 1} F_k^i \sum_{j \in \mathcal{N}_{i,k}^*} (\bar{\eta}_k^j - \bar{\eta}_k^i) \\ &= F_k^i \bar{\eta}_k^i + \left(\frac{1}{|\mathcal{N}_{i,k}^*| + 1} F_k^i \sum_{j \in \mathcal{N}_{i,k}^*} \bar{\eta}_k^j \right) - \frac{|\mathcal{N}_{i,k}^*|}{|\mathcal{N}_{i,k}^*| + 1} F_k^i \bar{\eta}_k^i \\ &= \frac{1}{|\mathcal{N}_{i,k}^*| + 1} F_k^i \sum_{j \in \mathcal{N}_{i,k}^* \cup \{i\}} \bar{\eta}_k^j \\ \bar{\eta}_{k+1}^i &= A \eta_k^i,\end{aligned}\tag{4.22}$$

and the augmented noiseless error dynamics are

$$\begin{aligned}\eta_k &= \text{diag}\{F_k^i A\}_{i=1}^N \left(I_{Nn} - \left(\mathcal{D}_k^{-1} L_k^* \otimes I_n \right) \right) \eta_{k-1} \\ &= \text{diag}\{F_k^i\}_{i=1}^N \left((I_N - \mathcal{D}_k^{-1} L_k^*) \otimes A \right) \eta_{k-1},\end{aligned}\tag{4.23}$$

with $\mathcal{D}_k = \text{diag}\{|\mathcal{N}_{i,k}^*| + 1\}_{i=1}^N$. We note that for the case where each agent constantly triggers, i.e., when $L_k^* = L_k \forall k$, we obtain the noiseless error dynamics (3.29). Additionally for the non-cooperative case, i.e., when $(I_N - \mathcal{D}_k^{-1} L_k^*) = I_N$, we obtain the noiseless NCLKF error dynamics.

Under the case where each sensor has the same observation of the process, we can arrive at the following result.

Proposition 4.2.1. *Assume that each sensor in the network measures the process (4.1) using the same observation model*

$$z_k^i = H x_k + v_k^i, i = 1, \dots, N,$$

where v_k^i is the zero-mean Gaussian measurement noise with $\mathbb{E}[v_k^i (v_l^i)^T] = R \delta_{kl}$. Additionally, assume that each agent activates the event-triggered consensus Kalman filter (4.19) with the consensus gain (4.20) and the event-triggered mechanism with event trigger condition (4.18). Then the error dynamics (4.23) are asymptotically stable.

Proof. This proof follows a similar line to the proof for Proposition 3.3.1. In the case where each sensor uses the same measurement model, it follows that $F_k^i = \bar{F}_k$ for all

agents. The error dynamics can then be simplified to

$$\begin{aligned}\eta_k &= \text{diag}\{F_k^i A\}_{i=1}^N \left(I_{Nn} - \left(\mathcal{D}_k^{-1} L_k^* \otimes I_n \right) \right) \eta_{k-1} \\ &= (I_N \otimes \bar{F}_k A) \left((I_N - \left(\mathcal{D}_k^{-1} L_k^* \right)) \otimes I_n \right) \eta_{k-1} \\ &= \left((I_N - \left(\mathcal{D}_k^{-1} L_k^* \right)) \otimes \bar{F}_k A \right) \eta_{k-1}.\end{aligned}$$

Due to the properties of the Kronecker product, we have that $(I_N \otimes \bar{F}_k A)$ and $((I_N - (\mathcal{D}_k^{-1} L_k^*)) \otimes I_n)$ commute. This leads to the following inequality,

$$\lim_{k \rightarrow \infty} \left\| \prod_k \left((I_N - (\mathcal{D}_k^{-1} L_k^*)) \otimes \bar{F}_k A \right) \right\| \leq \lim_{k \rightarrow \infty} \left\| \prod_k (\bar{F}_k A) \right\| \lim_{k \rightarrow \infty} \left\| \prod_k (I_N - (\mathcal{D}_k^{-1} L_k^*)) \right\|.$$

From the stability of the NCLKF, it follows that $\lim_{k \rightarrow \infty} \left(\prod_k \bar{F}_k A \right) = 0$. If ETC (4.18) is satisfied for all agents events we have that $(I_N - \mathcal{D}_k^{-1} L_k^*) = I_N$, if ETC (4.18) is violated for some agents, the Laplacian L_k^* will correspond with the new instantaneous topology and will not necessarily be symmetric. Finally, if ETC (4.18) is violated for all agents then $L_k^* = L_k$. For all mentioned scenarios, the matrix $(I_N - \mathcal{D}_k^{-1} L_k^*)$ is row stochastic at each time step k , and thus its spectral radius is always unity, and in particular,

$$\rho \left(\lim_{k \rightarrow \infty} \left(\prod_k (I_N - \mathcal{D}_k^{-1} L_k^*) \right) \right) = 1.$$

since the product of row-stochastic matrices are row-stochastic. Therefore,

$$\lim_{k \rightarrow \infty} \eta_k = \lim_{k \rightarrow \infty} \left(\prod_k (I_N - (\mathcal{D}_k^{-1} L_k^*)) \otimes \prod_k \bar{F}_k A \right) \eta_0 = 0.$$

and the noiseless error dynamics are asymptotically stable. ■

Remark 3. *For the proof of Proposition 3.3.1 we have assumed the sensing network communication graph is connected and undirected in each time step k . We now dismiss both of these assumption and find proof for the general case of disconnected and directed graphs.*

Remark 4. *In contrast to the event triggering mechanism discussed in Section 4.2.1, here the event trigger condition does not play a role in ruling stability on the error dynamics, instead, its contribution is with respect to performance.*

The result of Proposition 4.2.1, here as well, is restricted to the homogeneous case where each sensor has the same measurement noise characteristics. Although the proof for the stability of the heterogeneous case is not given in this research, we note that numerical simulations show promising results; we explore this in Section 4.4.

4.3 Stability with Partial Non-Observability

In this section we shall relax the assumption where all agents are able to measure the process (4.1). We consider the scenario where the sensing agents have a fixed sensing radius r , and a measurement is available only when the process state is within this radius. Formally, let $p^i \in \mathbb{R}^d$ be the fixed position of sensor i for $d \in \{2, 3\}$. Then the measurement model for each sensing agent now has the form

$$z_k^i = \begin{cases} Hx_k + v_k^i, & \|x_k - p^i\| \leq r, \\ \emptyset, & \text{otherwise} \end{cases}, \quad (4.24)$$

where $z_k^i = \emptyset$ means that there is no measurement available to agent i . This measurement model is illustrated in Figure 4.5a. Here, the process state is indicated by the black node. All the blue nodes are able to sense the process, while the red nodes are not. At the same time, the sensor agents are able to exchange information with each other according to the underlying communication graph, indicated by edges in the figure. Note that as the process state moves, the sensing agents may also change.

To cope with this scenario, we introduce some new notations to capture the state-dependent nature of the sensing network. First, we consider the network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of N sensing agents with an undirected communication topology (see Fig. 4.5a). Let L be the Laplacian matrix for the sensing network \mathcal{G} describing the available communication channels between each agent. Let $\mathcal{O}_k \subseteq \mathcal{V}$ denote the set of agents that are able to measure the process according to (4.24) at time-step k , and let $\bar{\mathcal{O}}_k$ be the set of agents that are not able to measure the process. Thus it holds that $\mathcal{O}_k \cup \bar{\mathcal{O}}_k = \mathcal{V}$ for all k . We define a directed network architecture to reflect the difference between the sensing and non-sensing nodes. This leads to a time varying and directed communication graph (see Fig. 4.5b) which is denoted by $\hat{\mathcal{G}}_k$. The edge set of $\hat{\mathcal{G}}_k$ is thus determined by which agents can observe the process. Non-observing nodes do not share any information with other nodes, but are able to receive information from the sensing nodes. Let

$$\begin{aligned} \mathcal{E}_{\mathcal{O}_k, \bar{\mathcal{O}}_k} &= \{(i, j) \mid i \in \mathcal{O}_k, j \in \bar{\mathcal{O}}_k, \{i, j\} \in \mathcal{E}\}, \\ \mathcal{E}_{\mathcal{O}_k, \mathcal{O}_k} &= \{(i, j) \mid i, j \in \mathcal{O}_k, \{i, j\} \in \mathcal{E}\}, \end{aligned} \quad (4.25)$$

be the set of edges connecting the observing agents to the non-observing agents, and observing agents with observing agents, respectively. Thus, $\hat{\mathcal{G}}_k = (\mathcal{V}, \mathcal{E}_{\mathcal{O}_k, \bar{\mathcal{O}}_k} \cup \mathcal{E}_{\mathcal{O}_k, \mathcal{O}_k}) = (\mathcal{V}, \hat{\mathcal{E}}_k)$. Note that the observing agents have bidirectional edges between them, while non-observing agents only have incoming edges emanating from the observing agents. This is illustrated in Fig. 4.5b. We make the following assumption on the time-varying structure of the graph $\hat{\mathcal{G}}_k$.

Assumption 2. *The graph $\hat{\mathcal{G}}_k$ is weakly connected at each time k . Equivalently, there*

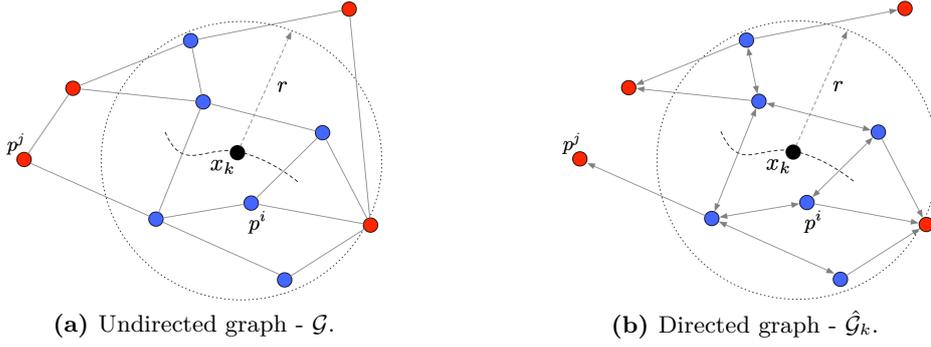


Figure 4.5: Measurement model where sensing agents can only sense the process when it is in range.

exists a directed path from every observing agent in \mathcal{O}_k to every non-observing agent in $\bar{\mathcal{O}}_k$, and $|\mathcal{O}_k| \geq 1$.

We now wish to extend our discussion by accounting for energy consumption in the form of communication loads. This shall be done for observing agents only, as non-observing agents shall not broadcast information according to the aforementioned architecture. This distinction is made clear using the following ETM,

$$\beta_k^j = \begin{cases} 1, & f_k^j(\{\tilde{x}_k^{sj}\}_{s \in \mathcal{N}_j \cup \{j\}}, \bar{x}_k^j) > 0 \\ 0, & f_k^j(\{\tilde{x}_k^{sj}\}_{s \in \mathcal{N}_j \cup \{j\}}, \bar{x}_k^j) \leq 0 \end{cases}, \quad (4.26)$$

where $\beta_k^j = 1$ means that the j th agent is broadcasting its state information and $\beta_k^j = 0$ means it does not broadcast. Additionally, the following ETF is used,

$$f_k^j(\{\tilde{x}_k^{sj}\}_{s \in \mathcal{N}_j \cup \{j\}}, \bar{x}_k^j) = \begin{cases} -1, & j \in \bar{\mathcal{O}}_k \\ 1, & j \in \cup_{i \in \bar{\mathcal{O}}_k} \mathcal{N}_i, j \in \mathcal{O}_k, \\ (\bar{x}_k^j - \tilde{x}_k^{jj})^T (\bar{x}_k^j - \tilde{x}_k^{jj}) - \delta, & j \notin \cup_{i \in \bar{\mathcal{O}}_k} \mathcal{N}_i, j \in \mathcal{O}_k \end{cases} \quad (4.27)$$

where \tilde{x}^{jj} is formulated as in (4.4), $\delta \in \mathbb{R}$ is some constant threshold and \mathcal{N}_i denote the neighborhood of the i th agent with respect to the sensing graph \mathcal{G} . The event-triggering function (4.27) is constructed such that non-observing agent will not broadcast their state estimation so that $\beta_k^j = 0 \forall j \in \bar{\mathcal{O}}_k$. Furthermore, observing agents which are neighbors to non-observing agents will constantly broadcast their state estimation so that $\beta_k^j = 1 \forall j \in \mathcal{O}_k \cap (\cup_{i \in \bar{\mathcal{O}}_k} \mathcal{N}_i)$. This ensures Assumption 2 holds. Finally, observing agents with no non-observing neighbors will broadcast their estimates based on the SoD event trigger condition so that β_k^j value is modified accordingly.

Let $\mathcal{G}_k^* = (\mathcal{V}, \mathcal{E}_k^*)$ denote the transmission graph of broadcasting agents at step k . The edge set of \mathcal{G}_k^* is thus determined by which agents are broadcasting such that

$\mathcal{E}_k^* = \{(j, i) \mid \beta_k^j = 1, j \in \mathcal{N}_i\}$. Additionally, let L_k^* be the Laplacian matrix for the directed sensing network \mathcal{G}_k^* .

To summarize, we have introduced three communication graphs: the available undirected communication graph \mathcal{G} (Fig. 4.5a), the weakly connected communication graph $\hat{\mathcal{G}}_k$ introduced to cope with partial non-observability (Fig. 4.5b), and the graph \mathcal{G}_k^* introduced to describe the instantaneous broadcasting nature of an event-triggered scheme (Fig. 4.4b). Note that $\mathcal{E}_k^* \subseteq \hat{\mathcal{E}}_k$.

We note that for this configuration, in order to ensure stability of the error dynamics, the consensus factor must be reevaluated. This is due to the fact that the noiseless error dynamic for agents with null observability are not necessarily asymptotically stable in the non-cooperative case. In fact, since in this scenario a non-observing agent is just propagating the state, we would expect to see the error diverge in some cases. Thus, the proof that was given in Section 4.2.2 is incompatible.

Additionally, the state estimator must be reconstructed as well. In this direction we consider the following event-triggered consensus Kalman estimator:

$$\hat{x}_k^i = \begin{cases} \bar{x}_k^i + K_k^i(z_k^i - H_k \bar{x}_k^i) + C_k^i \sum_{j \in \mathcal{N}_i} \beta_k^j (F_k^j \bar{x}_k^j - F_k^i \bar{x}_k^i), & i \in \mathcal{O}_k \\ \bar{x}_k^i + C_k^i \sum_{j \in \mathcal{N}_i} \beta_k^j (F_k^j \bar{x}_k^j - F_k^i \bar{x}_k^i), & i \in \bar{\mathcal{O}}_k \end{cases}. \quad (4.28)$$

where $F_k^i = I - K_k^i H$. Note that the innovation term nullifies for the non-observing case as no local measurements are obtained.

Now that our setup is complete, we may proceed to the stability analysis for the described configuration. Under the case where each observing agent has the same observation of the process, we can arrive at the following result.

Proposition 4.3.1 (Stability with Partial Observability). *Consider a group of N agents interacting over a time-varying graph $\hat{\mathcal{G}}_k = (\mathcal{V}, \mathcal{E}_{\mathcal{O}_k, \mathcal{O}_k} \cup \mathcal{E}_{\bar{\mathcal{O}}_k, \mathcal{O}_k})$ as defined in (4.25) and satisfying Assumption 2, where each agent observes the process (4.1) with the same state dependent observation model (4.24). Let the event-triggered consensus Kalman estimator be of type (4.28). Additionally, let the event-triggered mechanism be given as (4.26) with the event-triggered condition (4.27). Finally, let the consensus gain be given as*

$$C_k^i = \begin{cases} \frac{1}{1 + \sum_{j \in \mathcal{N}_i} \beta_k^j} & i \in \mathcal{O}_k \\ \frac{1}{\sum_{j \in \mathcal{N}_i} \beta_k^j} & i \in \bar{\mathcal{O}}_k \end{cases}. \quad (4.29)$$

Then the noiseless estimation error is asymptotically stable.

Proof. We begin our proof by constructing the error dynamics for the observing agents

and non-observing agents. Next we construct the joint error dynamics and prove that it is asymptotically stable.

In the case where each observing sensor uses the same measurement model, it follows that $F_k^i = \bar{F}_k \forall i \in \mathcal{O}_k$. Thus, the error dynamics for observing agents are:

$$\begin{aligned}
 \eta_k^i &= \bar{F}_k \bar{\eta}_k^i + \frac{1}{1 + \sum_{j \in \mathcal{N}_i} \beta_k^j} \bar{F}_k \sum_{j \in \mathcal{N}_i} \beta_k^j (\bar{\eta}_k^j - \bar{\eta}_k^i) \\
 &= \bar{F}_k \bar{\eta}_k^i + \left(\frac{1}{1 + \sum_{j \in \mathcal{N}_i} \beta_k^j} \bar{F}_k \sum_{j \in \mathcal{N}_i} \beta_k^j \bar{\eta}_k^j \right) - \frac{\sum_{j \in \mathcal{N}_i} \beta_k^j}{1 + \sum_{j \in \mathcal{N}_i} \beta_k^j} \bar{F}_k \bar{\eta}_k^i \\
 &= \frac{1}{1 + \sum_{j \in \mathcal{N}_i} \beta_k^j} \bar{F}_k \left(\bar{\eta}_k^i + \sum_{j \in \mathcal{N}_i} \beta_k^j \bar{\eta}_k^j \right) \\
 \bar{\eta}_{k+1}^i &= A \eta_k^i.
 \end{aligned} \tag{4.30}$$

Similarly, since the innovation is nullified for non-observing agents we have that $F_k^i = I_n \forall i \in \bar{\mathcal{O}}_k$ and so, the error dynamics for non observing agents are:

$$\begin{aligned}
 \bar{\eta}_k^i &= \bar{\eta}_k^i + \frac{1}{\sum_{j \in \mathcal{N}_i} \beta_k^j} \sum_{j \in \mathcal{N}_i} \beta_k^j (\bar{F}_k \bar{\eta}_k^j - \bar{\eta}_k^i) \\
 &= \bar{\eta}_k^i + \left(\frac{1}{\sum_{j \in \mathcal{N}_i} \beta_k^j} \bar{F}_k \sum_{j \in \mathcal{N}_i} \beta_k^j \bar{\eta}_k^j \right) - \bar{\eta}_k^i \\
 &= \frac{1}{\sum_{j \in \mathcal{N}_i} \beta_k^j} \bar{F}_k \sum_{j \in \mathcal{N}_i} \beta_k^j \bar{\eta}_k^j \\
 \bar{\eta}_{k+1}^i &= A \bar{\eta}_k^i,
 \end{aligned} \tag{4.31}$$

The joint error dynamics can now be constructed as such:

$$\begin{aligned}
 \eta_k &= \text{diag}\{F_k^i A\}_{i=1}^N \left(I_{Nn} - \left(\mathcal{D}_k^{-1} L_k^* \otimes I_n \right) \right) \eta_{k-1} \\
 &= (I_N \otimes \bar{F}_k A) \left((I_N - \left(\mathcal{D}_k^{-1} L_k^* \right)) \otimes I_n \right) \eta_{k-1} \\
 &= \left((I_N - \left(\mathcal{D}_k^{-1} L_k^* \right)) \otimes \bar{F}_k A \right) \eta_{k-1}.
 \end{aligned}$$

where $\mathcal{D}_k = \text{diag}\{d_k^i\}_{i=1}^N$ and

$$d_k^i = \begin{cases} 1 + \sum_{j \in \mathcal{N}_i} \beta_k^j & i \in \mathcal{O}_k \\ \sum_{j \in \mathcal{N}_i} \beta_k^j & i \in \bar{\mathcal{O}}_k \end{cases}.$$

By Assumption 2 we have that $d_k^i > 0$ for all agents, therefore the matrix \mathcal{D}_k is invertible. Due to the properties of the Kronecker product, we have that $(I_N \otimes \bar{F}_k A)$

and $((I_N - (\mathcal{D}_k^{-1}L_k^*)) \otimes I_n)$ commute. This leads to the following inequality,

$$\lim_{k \rightarrow \infty} \left\| \prod_k \left((I_N - (\mathcal{D}_k^{-1}L_k^*)) \otimes \bar{F}_k A \right) \right\| \leq \lim_{k \rightarrow \infty} \left\| \prod_k (\bar{F}_k A) \right\| \lim_{k \rightarrow \infty} \left\| \prod_k (I_N - (\mathcal{D}_k^{-1}L_k^*)) \right\|.$$

The stability of the observers' NCLKF is ensured such that $\lim_{k \rightarrow \infty} \left(\prod_k \bar{F}_k A \right) = 0$. Additionally, ETM (4.17) dictates the values of the instantaneous Laplacian L_k^* , however the row stochastic property of the matrix $(I_N - \mathcal{D}_k^{-1}L_k^*)$ is not affected. Thus, its spectral radius is always unity such that,

$$\rho \left(\lim_{k \rightarrow \infty} \left(\prod_k (I_N - \mathcal{D}_k^{-1}L_k^*) \right) \right) = 1,$$

since the product of row-stochastic matrices is row-stochastic. Therefore,

$$\lim_{k \rightarrow \infty} \eta_k = \lim_{k \rightarrow \infty} \left(\prod_k (I_N - (\mathcal{D}_k^{-1}L_k^*)) \otimes \prod_k \bar{F}_k A \right) \eta_0 = 0.$$

and the error dynamics are asymptotically stable. ■

We have shown that by using the tools that were developed so far, we may approach unique scenarios in which some of the agents do not observe the process, and still obtain stability of the error dynamics. We strengthen our claim with a numerical example in the next section.

4.4 Simulation Results

We consider the same numerical example that was discussed in chapter 3 where a group of 20 agents observe a robot performing a noisy “snail” trajectory with the following dynamics,

$$x_{k+1} = \underbrace{\begin{bmatrix} 0.9996 & -0.0283 \\ 0.0283 & 0.9996 \end{bmatrix}}_A x_k + \underbrace{0.375 \cdot I_2}_B w_k. \quad (4.32)$$

The robot's initial state is set to be $x_0 = [15, -10]^T$, the initial covariance matrix for each agent is set to be $P_0^i = 10I$, and the agents' initial estimates are normally distributed about the initial state. Additionally, the process noise covariance is $Q = I_2$. The sensors are randomly positioned in some field of interest (see Fig. 4.6) where a communication link between 2 sensors exists only if their distance is below some threshold (< 40 meters). Furthermore, we consider two sensing models: 1) the homogeneous sensing model where each agent measure the robot with the same observation model

such that $R^i = R = 9$ and $H^i = H = [0.5, 0.5]$, and 2) the heterogeneous model where each agent with an even number measures the robot's y -axis position while the agents with an odd number measure its x -axis position such that:

$$H^i = \begin{cases} [1, 0] & i \in \{1, 3, \dots, 19\} \\ [0, 1] & i \in \{2, 4, \dots, 20\} \end{cases}. \quad (4.33)$$

The measurement noise covariance for the i th agent is $R^i = \sqrt{i}$.

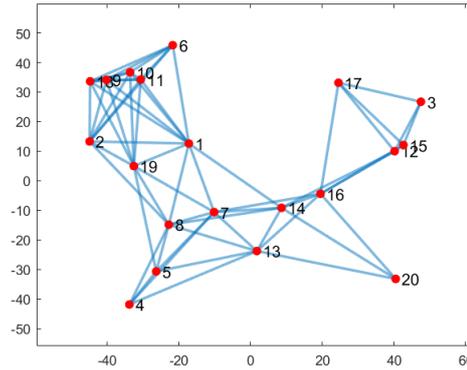


Figure 4.6: A sensor network of 20 agents randomly positioned.

We provide a comparison between 6 state estimators:

NCLKF: the non-cooperative local Kalman filter with null consensus gain;

CKF the sub-optimal consensus Kalman filter consensus gain (3.20) with the factor (4.14).

ETCKF1: the sub-optimal event triggered consensus Kalman filter with consensus gain (3.20), consensus factor (4.14), and event-triggered condition (4.10);

ETCKF2: the sub-optimal event triggered consensus Kalman filter with consensus gain (4.15) and event-triggered condition (4.10);

ETDCKF: the sub-optimal event-triggered decentralized consensus Kalman filter with event triggered condition (4.18), $\delta = 0.2$, and the consensus gain (4.20);

POETCKF: the sub-optimal event triggered decentralized consensus Kalman filter with event triggered condition (4.27), $\delta = 0.2$, and the consensus gain (4.29).

The performance of these estimators was tested over 100 Monte-Carlo simulations in which the process and measurement noises were randomized. The compared performance measures are twofold: the agents local energy consumption expressed as total number of events per agent, denoted by $\#^i$ (Fig. 4.8), calculated in the following manner:

$$\#^i = \frac{1}{MC} \sum_{j=1}^{MC} \#^{i,j},$$

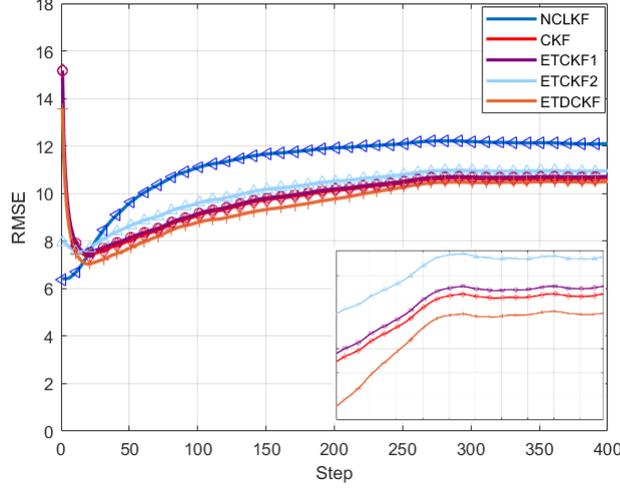


Figure 4.7: Root mean squared error, comparison between 5 state estimators over 100 Monte-Carlo runs with a homogeneous sensing model.

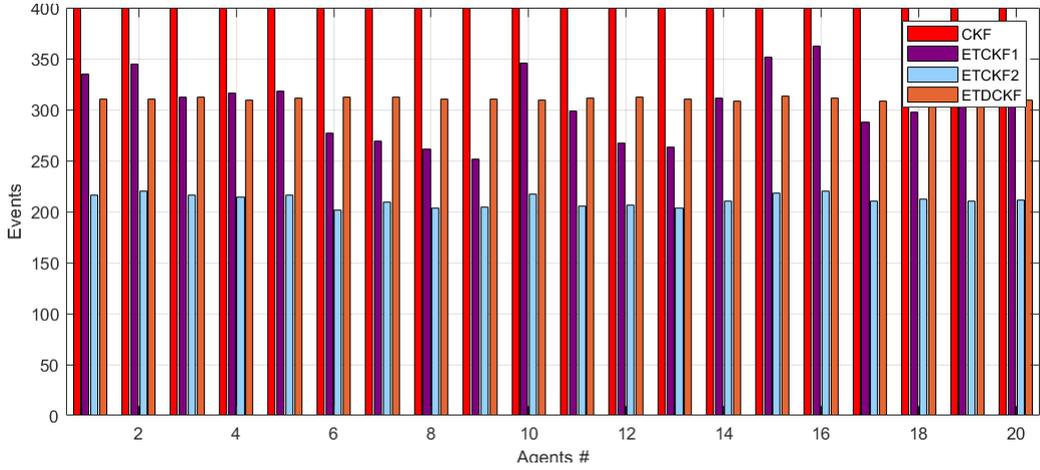


Figure 4.8: Total events per agent for a 400 step simulation - a comparison between 3 event triggered estimators and the CKF with a homogeneous sensing model.

where MC is the number of Monte-Carlo runs and $\#^{i,j}$ is the i th agent's total number of triggered events for the j th run. Additionally we have the true averaged root mean squared error (Fig. 4.7) calculated as such:

$$\text{RMSE} = \frac{1}{MC} \sum_{j=1}^{MC} \sqrt{\sum_{i=1}^N \mathbb{E}[(\eta^{i,j})^T \eta^{i,j}]},$$

where $\eta^{i,j}$ is the i th agent state estimation error for the j th run.

We begin our discussion with the homogeneous sensing model. Fig. 4.7 illustrates the averaged RMSE for 5 estimators. As shown, the ETDCKF is preferable in performance over the ETCKF1 and ETCKF2 and even over the CKF. This is explained

by the fact that for the ETDCKF, the consensus term is weighted according to the number of neighbors each agent has, while for the CKF all terms are weighted equally. Additionally, we see that the performance of the ETCKF1 is slightly better than that of ETCKF2. This result corresponds to the relatively low communication effort of this filter with respect to the others, as illustrated in Fig. 4.8. Overall, in this sensing scheme all estimators show superiority over the NCLKF. The performance of all estimators is summarized in Table 4.1 where we compare the averaged communication effort computed in the following manner, $\frac{\sum_{i=1}^N \#^i}{400N} \cdot 100\%$, where 400 is the simulation time.

Estimator	RMSE	Averaged com effort
NCLKF	12.2	0%
CKF	10.95	100%
ETCKF1	10.7	76%
ETCKF2	10.95	53%
ETDCKF	10.5	78%

Table 4.1: Estimators comparison for the homogeneous model.

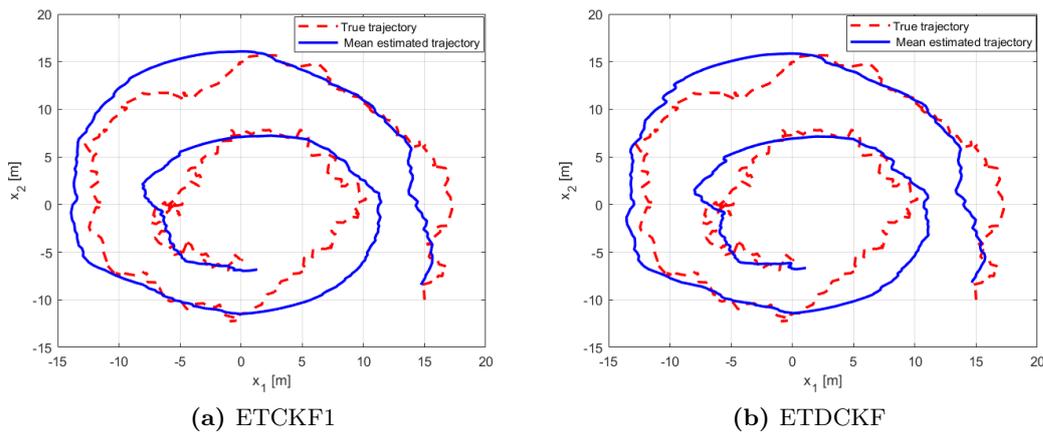


Figure 4.9: Trajectory of the true state and the agents' mean estimate utilizing ETCKF1 (a) and ETDCKF (b) for the homogeneous model.

The robot's true and mean estimated trajectory (using homogeneous sensing model) is depicted in Fig. 4.9, for a single run with both the ETCKF1 (4.9a) and the ETDCKF (4.9b). As shown, the filters provide reasonable tracking results.

Although we have proven the stability of ETDCKF for the case of homogeneous sensing model for all agents, we show in the following that in fact it provides reasonable results for the heterogeneous sensing model as well. Fig. 4.10 illustrates the averaged RMSE for the discussed 5 estimators. As shown, the effect of the consensus, with respect to the NCLKF, is much more dominant here compared to the homogeneous case. This is understandable since the flow of information, in this scheme, provides new "in-

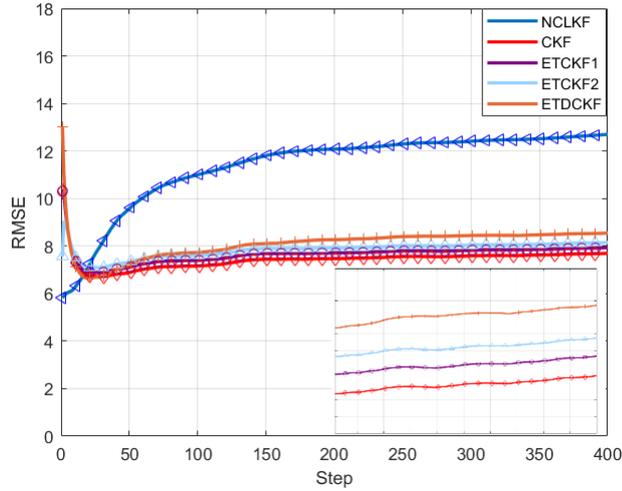


Figure 4.10: Root mean squared error, comparison between 5 state estimators over 100 Monte-Carlo runs with a heterogeneous sensing scheme.

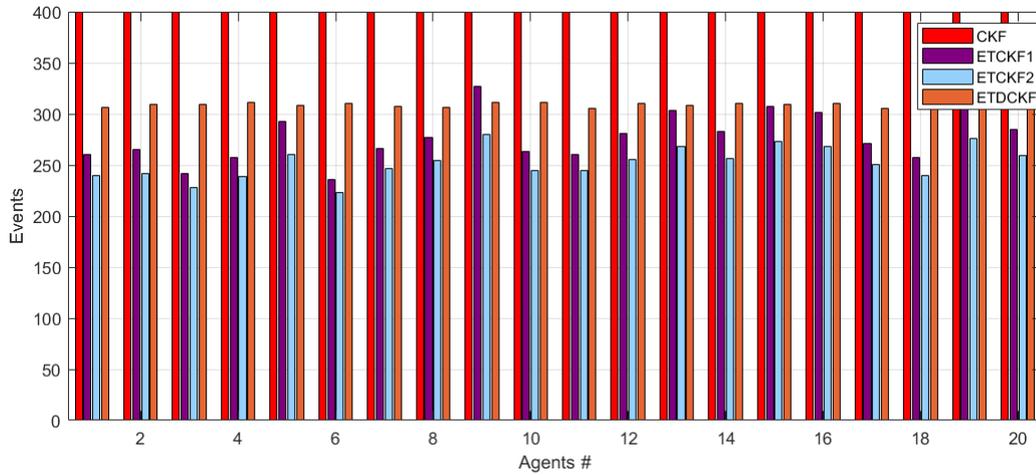


Figure 4.11: Total events per agent for a 400 step simulation - a comparison between 3 event triggered estimators and the CKF with a heterogeneous sensing scheme.

sights” for some agents. Specifically, ETCKF2 shows degraded performance compared to ETCKF1, and slightly better performance than ETDCKF. This too corresponds to the relatively low communication effort of this filter with respect to the others. This is illustrated in Fig. 4.11 where we see that estimators ETCKF1 and ETCKF2 are conservative while ETDCKF consumes a relative large amount of energy. What is interesting to see is that although the ETDCKF consumes more energy, its performance is degraded compared to ETCKF1, which indicates that the estimator structure has a dominant effect on performance for this scheme. The performance of all 5 estimators is summarized in Table 4.2.

The robot’s true and mean estimated trajectory (using the heterogeneous sensing model) is depicted in Fig. 4.12, for a single run with both the ETCKF1 (4.12a) and

Estimator	RMSE	Averaged com effort
NCLKF	12.7	0%
CKF	7.7	100%
ETCKF1	7.9	70%
ETCKF2	8.2	64%
ETDCKF	8.5	78%

Table 4.2: Estimators comparison for the heterogeneous model.

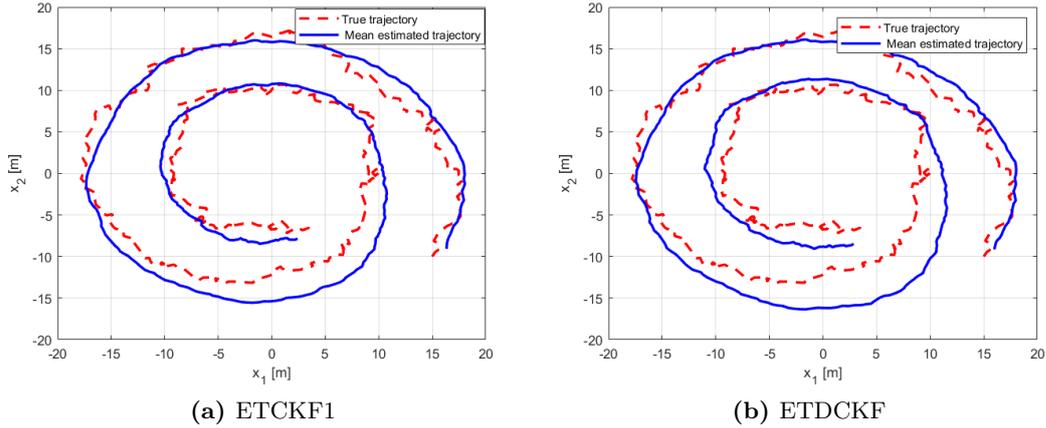


Figure 4.12: Trajectory of the true state and the agents' mean estimate utilizing ETCKF1 (a) and ETDCKF (b) for the heterogeneous model.

the ETDCKF (4.12b). As shown, the filter provides good tracking results.

To further demonstrate the robustness of the ETDCKF, we compared the sum of all agents MSE of a single run, with the heterogeneous sensing model, to that of the CKF and NCLKF. Only here, we have simulated an “unexpected” communication topology switch at two time instances, at $k = 50$ and $k = 150$ (see Fig. 4.13). In this scenario, the CKF estimator is maintaining the nominal communication topology of the Laplacian eigenvalues for calculating (4.14). Since they are no longer compatible after the first switch, this leads the CKF to become unstable immediately afterwards, while the ETDCKF remains stable for the entire duration.

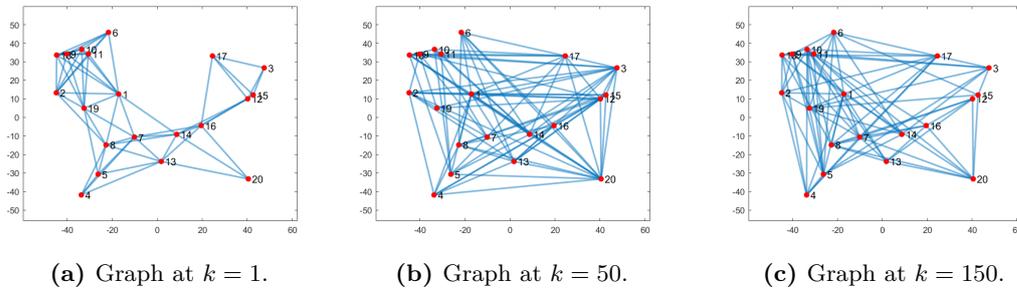


Figure 4.13: Communication graph at time steps (a) 1-49, (b) 50-149, and (c) 150-400.

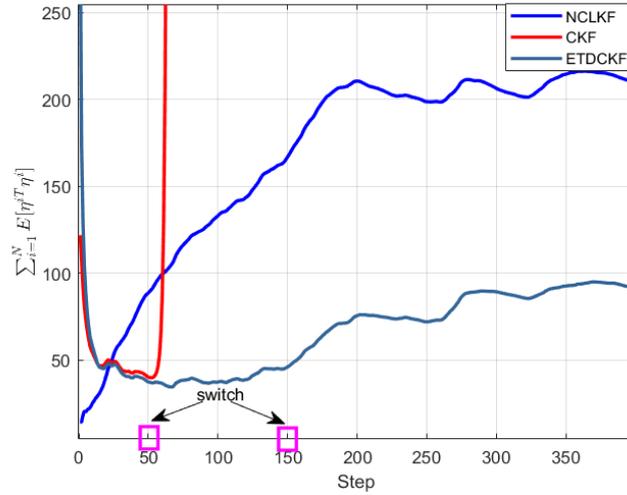


Figure 4.14: Sum of all agents mean squared error for 2 mid-run graph switch (at step 50 and at step 150), comparison between 3 state estimators for a single run.

To conclude, we investigate the performance of the scenario where some of the agents do not obtain any measurements. To do so, we simulate the limitation where the agents can not measure the state once the physical distance from the target is greater then some threshold. For the following analysis, we change the initial state vector of the robot to be $x_0 = [-20, -40]^T$. The sensors are randomly positioned in some field of interest (see Fig. 4.15) where a communication link between 2 sensors exists only if their distance is below 40 meters. Additionally we have simulated an out of range value of 85 m such that agents which their distance from the robot is above this value will not measure the robot's state according to observation rule (4.24). This is illustrated for two time instances during the run in Fig. 4.16 where non-observing agents are marked in red.

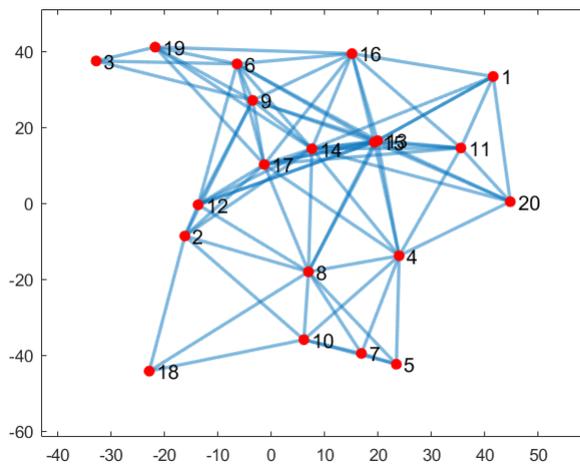


Figure 4.15: Available communication graph for a sensing network with 20 agents.

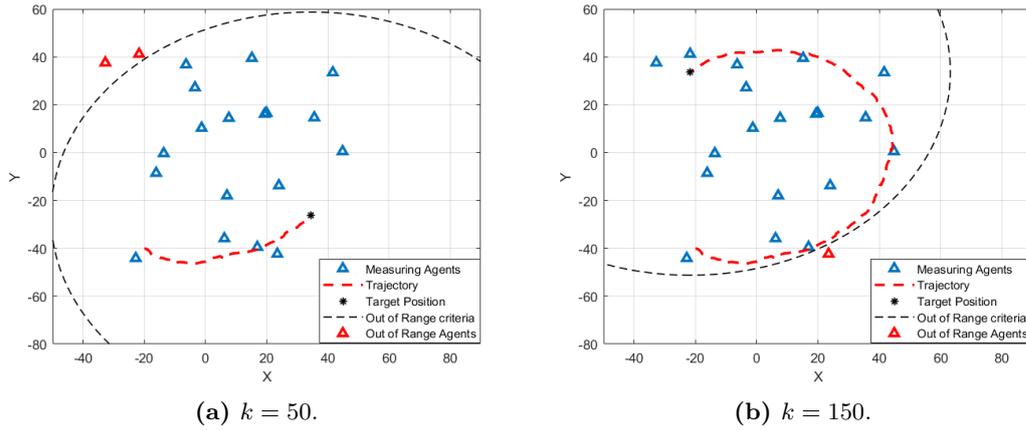


Figure 4.16: Observing and non-observing agents for an 85 meters out of range limitation at $k = 50$ (a) and $k = 150$ (b).

Fig. 4.17b compares the sum of all agents' MSE for 2 estimators: the NCLKF and POETCKF. As shown, the POETCKF provide reasonable results for a scenario with occasional partial non-observability. This figure also captures the disadvantage in running the non-cooperative estimator as the sum of MSE diverges since some of the agents' observability is lost. Furthermore, Fig 4.17a depicts the events map of all agents as a function of time. Here, the performance is satisfying with an averaged communication effort of 73%, which indicates that this scheme is effective.

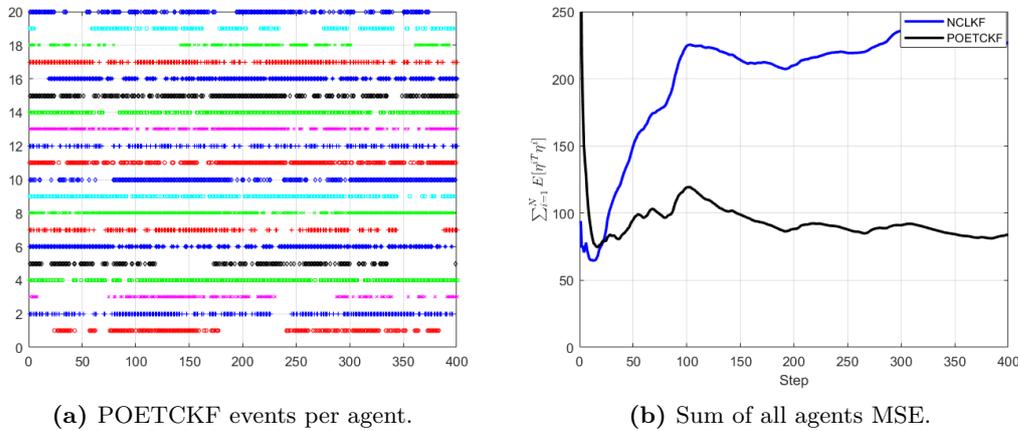


Figure 4.17: Partial observability scenario results.

To summarise this chapter, we have demonstrated superiority of our proposed centralized consensus gain compared to existing solutions and the non-cooperative Kalman estimator for both homogeneous and heterogeneous sensing models. We have demonstrated robustness to time varying communication topology for the decentralised consensus gain. Finally, we have shown an effective and simple architecture for scenarios in which some agents are locally non-observable.

Chapter 5

Conclusion and Open Questions

In this Chapter we conclude the work conducted for this thesis and also suggest future occupation following the line of our results.

5.1 Conclusion

In Chapter 3 we presented a widely common sub-optimal *consensus Kalman* filter scheme and presented new solutions for determining the consensus gain. In the centralized scheme, we proposed a semi-definite program for extracting an upper bound on the consensus factor which does not affect the state estimation error stability. Additionally, we proposed a decentralized scheme which does not require global knowledge of graph properties and can cope with varying communication topology. Finally we presented performance superiority of both schemes over existing solutions in the literature and over the non-cooperative local Kalman filter. In Chapter 4, we discussed a sub-optimal event-triggered consensus Kalman filter scheme and proposed new solutions for determining the consensus gain. In the centralized scheme, we utilized an existing event-triggered mechanism to conduct a full stability analysis for a unified consensus gain structure and our proposed centralized consensus gain. In the decentralized scheme, we proposed a decentralized consensus gain which does not require global knowledge of graph properties along with suggesting an event triggering scheme which will not affect stability. Additionally, we relaxed the assumption that all agents have full observability and constructed an event trigger architecture that deals with "blind" agents. Finally we have presented the effect of a low and high total event percentage with respect to performance by comparing our solution to both the non-cooperative local Kalman filter and the continuous consensus Kalman filter.

To conclude, in this work we have generated a toolbox (see Fig. 5.1) for cooperative estimation networks designers. After characterizing the network's requirements and con-

straints (energy consumption, access to global network properties, etc...), the designer may pinpoint the consensus Kalman estimator, consensus gain, and event-triggering mechanism that is most suitable for his or her specifications.

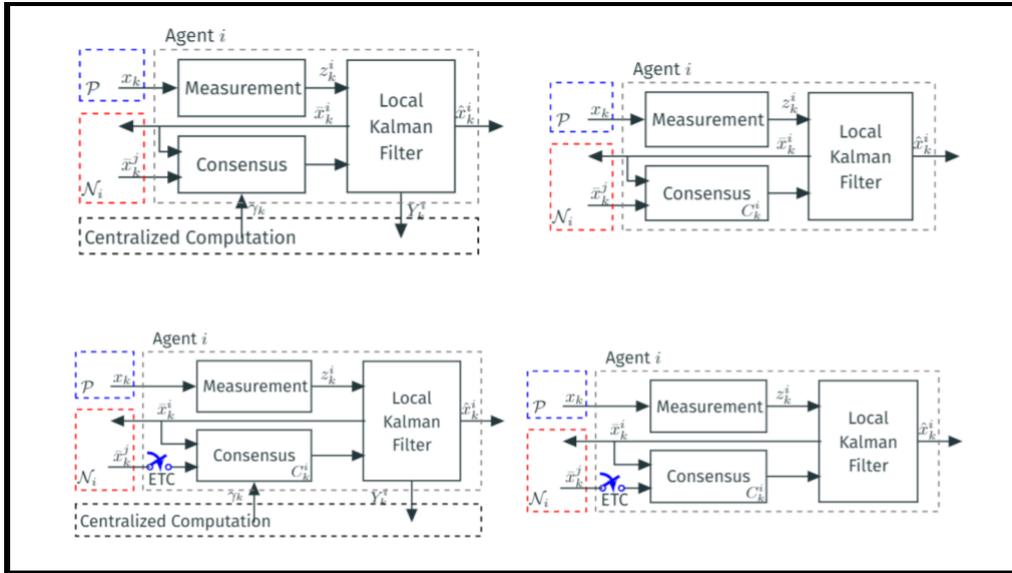


Figure 5.1: Cooperative estimation network designers toolbox.

5.2 Future Work

This thesis may serve as the building block to explore new ideas and to answer future questions in the field of distributed estimation and event-triggered estimation. To start with, future work can be devoted to projecting the methodologies that were used in this thesis onto other distributed estimators such as the EKF, the Unscented KF and more. Performance wise, one direction is to find the optimal event-triggered CKF, with respect to the MSE, and conduct comparisons with the event-triggered schemes presented in this work. In this direction, future researchers may consider redesign the cost function to include a penalty on transmission.

On the theoretical side, much work remains to be done. The properties of the estimators presented in this work, such as the biasness of the mean squared, can be explored. For example it is obvious that the estimate of the error covariance is biased due to the sub-optimal assumption that we have made, however what about the biasness of the estimate itself? Another direction is to explore an upper bound on the error covariance. To do so for the NCLKF case, [29] constructed the observation Gramian and extracted an upper bound on the error covariance. Similarly, this can be done for the cooperative case where the distributed network observation Gramian should be defined and used to construct an upper bound on the i th agent error covariance.

Finally, in this work we presented proof for the stability of the consensus Kalman estimator, which incorporates only local network properties, under the assumption of a homogeneous sensing model. This proof should be extended for the heterogeneous case as well.

Bibliography

- [1] Mohiuddin Ahmed, Yung-Szu Tu, and Gregory Pottie. Cooperative detection and communication in wireless sensor networks. In *38th Allerton Conf. On Comm. Control, and Computing*, pages 755–764, 2000.
- [2] Ian F Akyildiz, Dario Pompili, and Tommaso Melodia. Challenges for efficient communication in underwater acoustic sensor networks. *ACM Sigbed Review*, 1(2):3–8, 2004.
- [3] Ian F Akyildiz and Mehmet Can Vuran. *Wireless sensor networks*, volume 4. John Wiley & Sons, 2010.
- [4] Mehdi Alighanbari and Jonathan P How. Unbiased kalman consensus algorithm. *Journal of Aerospace Computing, Information, and Communication*, 5(9):298–311, 2008.
- [5] William N Anderson Jr and Thomas D Morley. Eigenvalues of the laplacian of a graph. *Linear and multilinear algebra*, 18(2):141–145, 1985.
- [6] Stefano Battilotti, Filippo Cacace, Massimiliano d’Angelo, and Alfredo Germani. Asymptotically optimal consensus-based distributed filtering of continuous-time linear systems. *Automatica*, 122:109189, 2020.
- [7] Brett Bethke, Mario Valenti, and Jonathan How. Cooperative vision based estimation and tracking using multiple uavs. In *Advances in cooperative control and optimization*, pages 179–189. Springer, 2007.
- [8] Raj Deshmukh, Cheolhyeon Kwon, and Inseok Hwang. Optimal discrete-time kalman consensus filter. In *2017 American Control Conference (ACC)*, pages 5801–5806. IEEE, 2017.
- [9] Francesco Fraternali. Towards large-scale autonomous wireless sensor networks. *arXiv preprint arXiv:1906.12001*, 2019.

- [10] Federica Garin and Luca Schenato. *A Survey on Distributed Estimation and Control Applications Using Linear Consensus Algorithms*, pages 75–107. Springer London, London, 2010.
- [11] Xiaohua Ge, Qing-Long Han, Xian-Ming Zhang, and Derui Ding. Dynamic event-triggered control and estimation: a survey. *International Journal of Automation and Computing*, pages 1–30, 2021.
- [12] Xiaohua Ge, Qing-Long Han, Xian-Ming Zhang, Lei Ding, and Fuwen Yang. Distributed event-triggered estimation over sensor networks: A survey. *IEEE transactions on cybernetics*, 50(3):1306–1320, 2019.
- [13] George T Gilbert. Positive definite matrices and sylvester’s criterion. *The American Mathematical Monthly*, 98(1):44–46, 1991.
- [14] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. \ urlhttp://cvxr.com/cvx, March 2014.
- [15] Fei Han, Yan Song, Sunjie Zhang, and Wangyan Li. Local condition-based finite-horizon distributed h_∞ -consensus filtering for random parameter system with event-triggering protocols. *Neurocomputing*, 219:221–231, 2017.
- [16] Ondrej Hlinka, Franz Hlawatsch, and Petar M Djurić. Consensus-based distributed particle filtering with distributed proposal adaptation. *IEEE Transactions on Signal Processing*, 62(12):3029–3041, 2014.
- [17] Ali Jadbabaie, Jie Lin, and A Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on automatic control*, 48(6):988–1001, 2003.
- [18] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.
- [19] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [20] Michael Lemmon. Event-triggered feedback in control, estimation, and optimization. In *Networked control systems*, pages 293–358. Springer, 2010.
- [21] Wenling Li, Yingmin Jia, and Junping Du. Event-triggered kalman consensus filter over sensor networks. *IET Control Theory & Applications*, 10(1):103–110, 2016.
- [22] Yuan Liang, Yinya Li, Sujuan Chen, Guoqing Qi, and Andong Sheng. Event-triggered kalman consensus filter for sensor networks with intermittent observations. *International Journal of Adaptive Control and Signal Processing*, 2021.

- [23] Qinyuan Liu, Zidong Wang, Xiao He, and DongHua Zhou. A survey of event-based strategies on control and estimation. *Systems Science & Control Engineering: An Open Access Journal*, 2(1):90–97, 2014.
- [24] Joel George Manathara and Debasish Ghose. Rendezvous of multiple uavs with collision avoidance using consensus. *Journal of Aerospace Engineering*, 25(4):480–489, 2012.
- [25] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. 1979.
- [26] Xiangyu Meng and Tongwen Chen. Optimality and stability of event triggered consensus state estimation for wireless sensor networks. In *2014 American Control Conference*, pages 3565–3570. IEEE, 2014.
- [27] Aleksandar Milenković, Chris Otto, and Emil Jovanov. Wireless sensor networks for personal health monitoring: Issues and an implementation. *Computer communications*, 29(13-14):2521–2533, 2006.
- [28] Adam Molin and Sandra Hirche. An iterative algorithm for optimal event-triggered estimation. *IFAC Proceedings Volumes*, 45(9):64–69, 2012.
- [29] JB Moore and Brian DO Anderson. Coping with singular transition matrices in estimation and control stability theory. *International Journal of Control*, 31(3):571–586, 1980.
- [30] Tamoghna Ojha, Sudip Misra, and Narendra Singh Raghuwanshi. Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. *Computers and electronics in agriculture*, 118:66–84, 2015.
- [31] 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.
- [32] Reza Olfati-Saber. Distributed kalman filtering for sensor networks. In *2007 46th IEEE Conference on Decision and Control*, pages 5492–5498. IEEE, 2007.
- [33] Reza Olfati-Saber. 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*, pages 7036–7042. IEEE, 2009.
- [34] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

- [35] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, 49(9):1520–1533, 2004.
- [36] Ertan Onur, Cem Ersoy, Hakan Deliç, and Lale Akarun. Surveillance wireless sensor networks: Deployment quality analysis. *IEEE Network*, 21(6):48–53, 2007.
- [37] Wei Ren. Consensus strategies for cooperative control of vehicle formations. *IET Control Theory & Applications*, 1(2):505–512, 2007.
- [38] Alejandro Ribeiro and Georgios B Giannakis. Bandwidth-constrained distributed estimation for wireless sensor networks-part i: Gaussian case. *IEEE transactions on signal processing*, 54(3):1131–1143, 2006.
- [39] Nils F Sandell and Reza Olfati-Saber. Distributed data association for multi-target tracking in sensor networks. In *2008 47th IEEE Conference on Decision and Control*, pages 1085–1090. IEEE, 2008.
- [40] Joongbo Seo, Youdan Kim, Seungkeun Kim, and Antonios Tsourdos. Consensus-based reconfigurable controller design for unmanned aerial vehicle formation flight. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 226(7):817–829, 2012.
- [41] RK Singh and Lyes Benyoucef. A consensus based group decision making methodology for strategic selection problems of supply chain coordination. *Engineering Applications of Artificial Intelligence*, 26(1):122–134, 2013.
- [42] Harold W Sorenson. Least-squares estimation: from gauss to kalman. *IEEE spectrum*, 7(7):63–68, 1970.
- [43] Tony Sun, Ling-Jyh Chen, Chih-Chieh Han, and Mario Gerla. Reliable sensor networks for planet exploration. In *Proceedings. 2005 IEEE Networking, Sensing and Control, 2005.*, pages 816–821. IEEE, 2005.
- [44] Valery Ugrinovskii. Distributed robust filtering with h_∞ consensus of estimates. *Automatica*, 47(1):1–13, 2011.
- [45] Jeremy G VanAntwerp and Richard D Braatz. A tutorial on linear and bilinear matrix inequalities. *Journal of process control*, 10(4):363–385, 2000.
- [46] Jingwei Wang, Eric A Butcher, and Yucelen Tansel. Space-based relative orbit estimation using information sharing and the consensus kalman filter. *Journal of Guidance, Control, and Dynamics*, 42(3):491–507, 2019.
- [47] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical Report 95-041, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

- [48] Li Wenshuang, Zhu Shanying, Chen Cailian, and Guan Xiping. Distributed consensus filtering based on event-driven transmission for wireless sensor networks. In *Proceedings of the 31st Chinese Control Conference*, pages 6588–6593. IEEE, 2012.
- [49] D Willner, CB Chang, and KP Dunn. Kalman filter algorithms for a multi-sensor system. In *1976 IEEE conference on decision and control including the 15th symposium on adaptive processes*, pages 570–574. IEEE, 1976.
- [50] Jacob Wolfowitz. Products of indecomposable, aperiodic, stochastic matrices. *Proceedings of the American Mathematical Society*, 14(5):733–737, 1963.
- [51] Anthony Wood, Gilles Virone, Thao Doan, Quihua Cao, Leo Selavo, Yafeng Wu, L Fang, Zhimin He, Shan Lin, and Jack Stankovic. Alarm-net: Wireless sensor networks for assisted-living and residential monitoring. *University of Virginia Computer Science Department Technical Report*, 2:17, 2006.
- [52] Max A Woodbury. *Inverting modified matrices*. Statistical Research Group, 1950.
- [53] Nan Xia, Fuwen Yang, and Qing-Long Han. Distributed event-triggered networked set-membership filtering with partial information transmission. *IET Control Theory & Applications*, 11(2):155–163, 2017.
- [54] Cui Zhang and Yingmin Jia. Distributed kalman consensus filter with event-triggered communication: formulation and stability analysis. *Journal of the Franklin Institute*, 354(13):5486–5502, 2017.
- [55] Hai-Ying Zhou, Dan-Yan Luo, Yan Gao, and De-Cheng Zuo. Modeling of node energy consumption for wireless sensor networks. *Wireless Sensor Network*, 3(1):18, 2011.

קודם ושדרוגם. בנוסף אנו מציעים הגבר קונצנזוס לא אחיד, כאשר גם במקרה זה מוצגת עמידות בפני שינויים במאפייני הרשת. אנו מראים כי מסנן זה הידוע בתור: מסנן קונצנזוס - קלמן מעורר אירועים, עולה בביצועיו על המסנן קלמן הקלאסי מאידך, ומחד חוסך בצורה משמעותית תעבורת תקשורת שוטפת כפי במבוצע במסנן קלמן - קונצנזוס הרציף. לסיום אנו מציעים ארכיטקטורת עירור אירועים בתנאים בהם חלק מהסוכנים ברשת איבדו את יכולת המדידה שלהם בצורה זמנית או תמידית. בנוסף אנו מספקים דוגמא נומרית להמחיש את יעילות תוצאותינו אל מול פתרונות הקיימים בספרות.

תקציר

מערכות חישה מרושתות מורכבות ממספר סוכנים בעלי יכולות תקשורת פנימית וחיישנים המאפשרים לבצע שיערוך שיתופי של איזשהו הליך פיזיקלי. בקונפיגורציה זו, כל סוכן מפעיל, באופן מבוזר, משערך המסתמך על מדידות מקומיות רועשות בהיתוך עם שיערוך המצב של סוכנים אחרים ברשת. שיערוך זה מתקבל ע"י תעבורת נתונים בין סוכנים. בבעיית השיערוך השיתופי, המערכת שואפת להתכנס גלובלית למצב האמיתי, תוך כדי התחשבות באלמנטים כגון: עומסי חישוב, כמות העברת נתונים וביצועים של כל סוכן בפרט ושל סך הסוכנים ככלל. פיתוח חדשני יחסית שנותן מענה לבעיית השיערוך השיתופי, הינו השילוב של רכיב קונצנזוס, מוכפל בהגבר מסויים, במסגן קלמן קלאסי. בארכיטקטורה זו כל סוכן מהתך מידע ממדידות מקומיות ביחד עם מידע מסוכנים אחרים ברשת עם יש לו תקשורת (שכנים). שיטת שיערוך זו ידועה בתור שיערוך קלמן - קונצנזוס. הגבר הקונצנזוס קובע את המשקל ששם הסוכן על שיערוך שכניו לעומת כמה הוא נשען על מדידותיו. כמובן שככל שהגבר קונצנזוס גדול, כך המשערך יסתמך יותר על שכניו והסכמה בין הסוכנים תושג יותר מהר (בהנחה שמשערך יציב). ניתן למצוא את הגבר הקונצנזוס האופטימלי על ידי גזירה של תוחלת ריבוע שגיאת השיערוך לפי ההגבר ולהשוותה לאפס. התוצאה שתתקבל תכלול ביטויים של שכנים מדרגה שניה, המשמעות היא שבעדכון משוואת המשערך כל סוכן ידרש לקבל מידע לא רק משכניו, אלא מהשכנים של שכניו. משיקולים פרקטים הרוב הגורף של המחקרים בנושא עוסק בפתרון תת - אופטימלי בו מגבילים את תעבורת המידע לשכנים דרגה ראשונה. תרומתינו מתחילה בהצעת הגבר קונצנזוס אחיד לכל הסוכנים אשר מחושב בשיטה של Programming Semi-Definite, בשיטה זו אנו מוצאים את הגבר הקונצנזוס המקסימלי עבורו כלל המשערכים ברשת יציבים. בכך, אנו מציגים שיפור בביצועים לעומת פתרונות אחרים הקיימים בספרות. יתרה מזאת, אנו מציעים הגבר קונצנזוס שאינו אחיד לכל הסוכנים, ומחושב מקומית, על ידי כל סוכן בנפרד, רק באמצעות מידע מקומי. בשיטה זו, ניתן לחשב את הגבר הקונצנזוס באופן מקוון, ללא הזרמה של מידע מיחידה חיצונית. דבר זה מוביל לחיסכון במשאבים, באנרגיה ובסיבוכיות. כמו כן המערכת אינה מתבססת על מאפייני רשת גלובלים ומציגה עמידות בפני שינויים במבנה הרשת בניגוד למקרה האחיד בו כיוול מחדש נדרש עבור כל שינוי מבנה רשת. אנו מרחיבים את המחקר בנושא מערכות חישה מרושתות על מנת לגעת בתחום חשוב ומדובר בספרות המודרנית - חיסכון באנרגיה וברזולוציה נמוכה יותר, חיסכון בתעבורת נתונים. אנו עושים זאת על ידי מימוש מנגנון עירור אירועים אשר מאפשר תעבורת נתונים רק כאשר תנאים מסויימים מתקיימים. תנאים אלו נבדקים ע"י כל סוכן בכל צעד זמן. תפקידו של מנגנון העירור הינו להכריע את נקודת הזמן בה יש צורך לבצע עדכון למידע המגיע מהסביבה. האתגר העיקרי במשערכים מסוג זה הינו התכן של הגבר הקונצנזוס ושל תנאי מעורר אירועים כך שדינמיקת השגיאה של כל סוכן תהיה יציבה, אך המערכת תשמור על רמת ביצועים נדרשת. בהמשך למוזכר, תרומתינו נמשכת בלקיחת המבנה של הגבר קונצנזוס אחיד ושל תנאי מעורר אירועים ממחקר

תודות

ראשית אני מבקש להודות לשותפתי לחיים, אשתי היקרה, שהסתדרה בגבורה לבדה ברגעים שהייתי צריך שקט לקדם את המחקר ובשבתות בהם לא הייתי נוכח.

שנית, אני מבקש לבטא את הערכתי ואת הוקרת תודתי למנחה שלי, פרופסור דניאל זלזו, על תמיכתו העיקשת ועל אמונתו בי. מילותיו שימשו לי כמקור אור מנחה בשדות לא מוכרים של תיאוריה ופרוצדורות אקדמיות. הוא נתן לי השראה בדרכים שתישארנה איתי למשך כל הקריירה המקצועית והאקדמאית שלי.

לצד המנחה, מבקש אני להודות לשאר חברי ועדת הבחינה שלי: פרופסור ליאוניד מירקין ופרופסור יעקב אושמן עבור זמנם, היערום החשובות ובעבור השיחה המעניינת שניהלנו.

בנוסף אני מבקש להודות לטכניון על תרומתו הנדיבה למימון המחקר.

לבסוף אני מעוניין להודות לקרן לי קא שינג על תרומתם הנדיבה למימון המחקר.

המחקר בוצע בהנחייתו של פרופסור דניאל זלזו, בפקולטה לאווירונאוטיקה וחלל.

חלק מן התוצאות בחיבור זה פורסמו כמאמרים מאת המחבר ושותפיו למחקר בכנסים ובכתבי-עת במהלך תקופת מחקר התזה של המחבר, אשר גרסאותיהם העדכניות ביותר הינן:

Aviv Priel and Daniel Zelazo. An improved distributed consensus Kalman filter design approach. In *2021 60th IEEE conference on Decision and Control*. IEEE, 2021.

מסנן קלמן עם קונצנזוס : תכנן המסנן ולוגיקת עירור לפי אירוע.

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים באווירונאוטיקה וחלל

אביב פריאל

הוגש לסנט הטכניון – מכון טכנולוגי לישראל
אדר א' התשפ"ב חיפה פברואר 2022

מסנן קלמן עם קונצנזוס : תכנן המסנן ולוגיקת עירור לפי אירוע.

אביב פריאל