FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of master's thesis

Title:	Distributed Kalman filtering under partially heterogeneous models
Student:	Bc. Thanh Quang Mai
Supervisor:	Ing. Kamil Dedecius, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2021/2022

Instructions

The state-of-the-art methods for distributed Kalman filtering mostly assume spatial homogeneity of the state-space models. However, in practice, the distributed network nodes may employ more or less different models. The thesis focuses on this case. In particular, the nodes observe a noisy realization of a 2D trajectory of a moving target and model it using either (i) the random-walk model, (ii) the constant velocity model, or (iii) the constant acceleration model. Naturally, the more complex models may work significantly better. The goal of the thesis is to propose a method improving the estimation performance and stability under the less complex (underspecified) models by collaboration among the nodes.

Electronically approved by Ing. Karel Klouda, Ph.D. on 6 January 2021 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

Distributed Kalman filtering under partially heterogeneous models

Bc. Thanh Quang Mai

Department of Applied Mathematics
Supervisor: Ing. Kamil Dedecius, Ph.D.

June 27, 2021

Acknowledgements

I would like to to thank my supervisor Ing. Kamil Dedecius, Ph.D., for his help, guidance and a great amount of patience.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on June 27, 2021 . . . . . .

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Mai, Thanh Quang. *Distributed Kalman filtering under partially heterogeneous models*. Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2021.

Abstrakt

Tato práce se zabývá problémem distribuovaného Kalmanovského filtrování při částečně heterogenních modelech. Je navrhnuta modifikace existujícího difuzního Kalmanova filtru, umožňující v difuzních sítích použití částečně heterogenních modelů. Výkon méně komplexních modelů je také zvýšen implementací heuristiky umožňující detekci selhávajících uzlů sítě, selhávající uzly jsou restartovány a je jim dána šance se zotavit.

Klíčová slova difuzní síť, difuzní odhadování, distribuované odhadování, odhadování stavu, Kalmanův filtr

Abstract

This thesis explores the problem of distributed Kalman filtering under partially heterogeneous models. A modification to the existing diffusion Kalman filter is proposed, enabling the employment of partially heterogeneous models in the diffusion networks. The performance of the less complex models is futher improved by the implementation of a node failure detection heuristic, resetting the failling nodes, and giving them a chance at a recovery.

Keywords diffusion network, diffusion estimation, distributed estimation, state estimation, Kalman filter

Contents

List of Figures

Introduction

Recently, advances in networked communication and the increasing availability of low-cost sensors have drawn an enormous amount of research attention to the field of distributed estimation. Its wide range of applications include tracking, navigation, medicine, telecommunications, monitoring, and so on. There are three main distributed strategies: incremental, consensus, and diffusion. In this thesis, the diffusion strategy will be of particular interest. In a diffusion network, the nodes only communicate with their adjacent neighbors, i. e., neighbors within a one-hop distance. The domain of distributed estimation of state-space models has mostly been dominated by the distributed versions of the Kalman filter [1, 2, 3, 4]. However, they mostly assume spatial homogeneity of the state-space models. This thesis focuses on the case of partial heterogeneity, where the network may employ models of various levels of complexity, and the less complex models are submodels of the more complex models. The goal of the thesis is to propose a collaboration method among the nodes, which would improve the estimation performance and stability of the less complex models in the network.

The thesis is organized as follows: Chapter 1 covers the preliminary theory of sequential estimation of state-space models. Chapter 2 follows by exploring the available distributed schemes, but then focusing exclusive on the adapt-then-combine diffusion strategy. Chapter 3 is concerned with the estimation of linear Gaussian state-space models — leading to the introduction of the Kalman filter and its distributed diffusion counterpart. Chapter 4 proposes a modification to the diffusion Kalman filter, enabling collaboration between nodes employing models of different levels of complexity. In addition, a simple node failure detection heuristic is proposed to further improve the estimation performance of the less complex models. Finally, Chapter 5 tests the performance of the proposed methods on a couple of simulated examples.

Chapter 1

Preliminaries

This chapter introduces the reader to the topic of discrete-time state-space models and the Bayesian approach to their sequential estimation. The linear model integral to this thesis is shown, and a few basic example models from the domain of target tracking are given. Finally, the Bayesian approach to estimation of models belonging to the exponential family is presented.

1.1 State-space model

Consider a discrete-time dynamical system with the latent state x_t and an observable output y_t, one way to describe such a system is state-space representation:

$$\begin{aligned} x_t &= f(x_{t-1}, u_t, w_t), \\ y_t &= g(x_t, v_t), \end{aligned} \tag{1.1}$$

where f and g are known functions, u_t — if it exists — is a known input (control) variable, w_t and v_t are zero-centered mutually independent noise variables [3].

Remark. (1.1) is a hidden Markov model (HMM). The hidden state x_t is not directly observable and can only be observed through the noisy output y_t.

1.2 Linear state-space model

A linear state-space model is a model in the following form:

$$\begin{aligned} x_t &= A_t x_{t-1} + B_t u_t + w_t, \\ y_t &= H_t x_t + v_t, \end{aligned} \tag{1.2}$$

where A_t, B_t, and H_t are known matrices of compatible dimensions. Noises w_t and v_t are zero-mean, mutually independent and identically distributed.

The linear system is time-invariant if it is described by the above equation and $A_t = A$, $B_t = B$, and $H_t = H$, i. e., the matrices A_t, B_t and H_t do not vary in time.

1.3 Kinematic models

In this section, the random-walk model (RWM), constant velocity model (CVM) and constant acceleration model (CAM) are introduced. CVM and CAM are widely used models derived from the physical equations of motion. A typical area of application is target tracking, e. g., tracking a moving object using a noisy remote sensor or tracking a moving object in a video.

The models are introduced using the time-invariant linear state-space model form (see (1.2)) with observable position and unknown or no input Bu_t, i.e.,

$$x_t = Ax_{t-1} + w_t, \tag{1.3}$$

$$y_t = Hx_t + v_t. \tag{1.4}$$

Since this thesis focuses on 2D tracking, the models are presented for two Cartesian coordinates and independency across coordinates is assumed. In addition, let us denote $\mathcal{WN}(0, M)$ a zero-mean white noise process with covariance matrix M.

The derivation of the below models can be found in [5, Sec. 6.1 – 6.3]. More complex mathematical models used for target tracking are described in [6].

1.3.1 Random-walk model

In the RWM the changes in position $\begin{bmatrix} x_{1,t} & x_{2,t} \end{bmatrix}^\mathsf{T}$ at each time t are modeled as a zero-mean white noise. The equations for target dynamics of the RWM are

$$\begin{aligned} x_{1,t} &= x_{1,t-1} + w_{x_1,t}, \\ x_{2,t} &= x_{2,t-1} + w_{x_2,t}. \end{aligned} \tag{1.5}$$

In matrix form (1.3), we have the state vector

$$x_t = \begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix},$$

with the state transition matrix

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

w_t is an independent process noise

$$w_t = \begin{bmatrix} w_{x_1,t} \\ w_{x_2,t} \end{bmatrix}, \quad w_t \sim \mathcal{WN}(0, Q),$$

with covariance matrix

$$Q = \begin{bmatrix} \Delta_t & 0 \\ 0 & \Delta_t \end{bmatrix} \widetilde{q},$$

where Δ_t is sampling period and $\widetilde{q}$ is process noise intensity. The observation equation is given by (1.4) with the observation matrix

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The observation noise v_t is an independent, zero-mean variable

$$v_t \sim \mathcal{WN}(0, R_t), \quad R_t \in \mathbb{R}^{2\times 2}.$$

An example of a RWM simulated trajectory can be seen in Figure 1.1.

1.3.2 Constant velocity model

This model assumes constant velocity in successive measurements. Changes in velocity (acceleration) are modeled as a zero-mean white noise. The state vector has four elements: $x_{1,t}$ and $x_{2,t}$ for target location and $v_{x_1,t}$, $v_{x_2,t}$ for target velocity. The corresponding difference equations are

$$\begin{aligned} x_{1,t} &= x_{1,t-1} + v_{x_1,t}\Delta_t + w_{x_1,t}, \\ x_{2,t} &= x_{2,t-1} + v_{x_2,t}\Delta_t + w_{x_2,t}, \\ v_{x_1,t} &= v_{x_1,t-1} + w_{v_{x_1},t}, \\ v_{x_2,t} &= v_{x_2,t-1} + w_{v_{x_2},t}. \end{aligned} \tag{1.6}$$

The equations written in matrix form (1.3), the state vector x_t is given by

$$x_t = \begin{bmatrix} x_{1,t} \\ x_{2,t} \\ v_{x_1,t} \\ v_{x_2,t} \end{bmatrix},$$

with the state transition matrix

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

w_t is an independent process noise

$$w_t = \begin{bmatrix} w_{x_1,t} \\ w_{x_2,t} \\ w_{v_{x_1},t} \\ w_{v_{x_2},t} \end{bmatrix}, \quad w_t \sim \mathcal{WN}(0, Q),$$

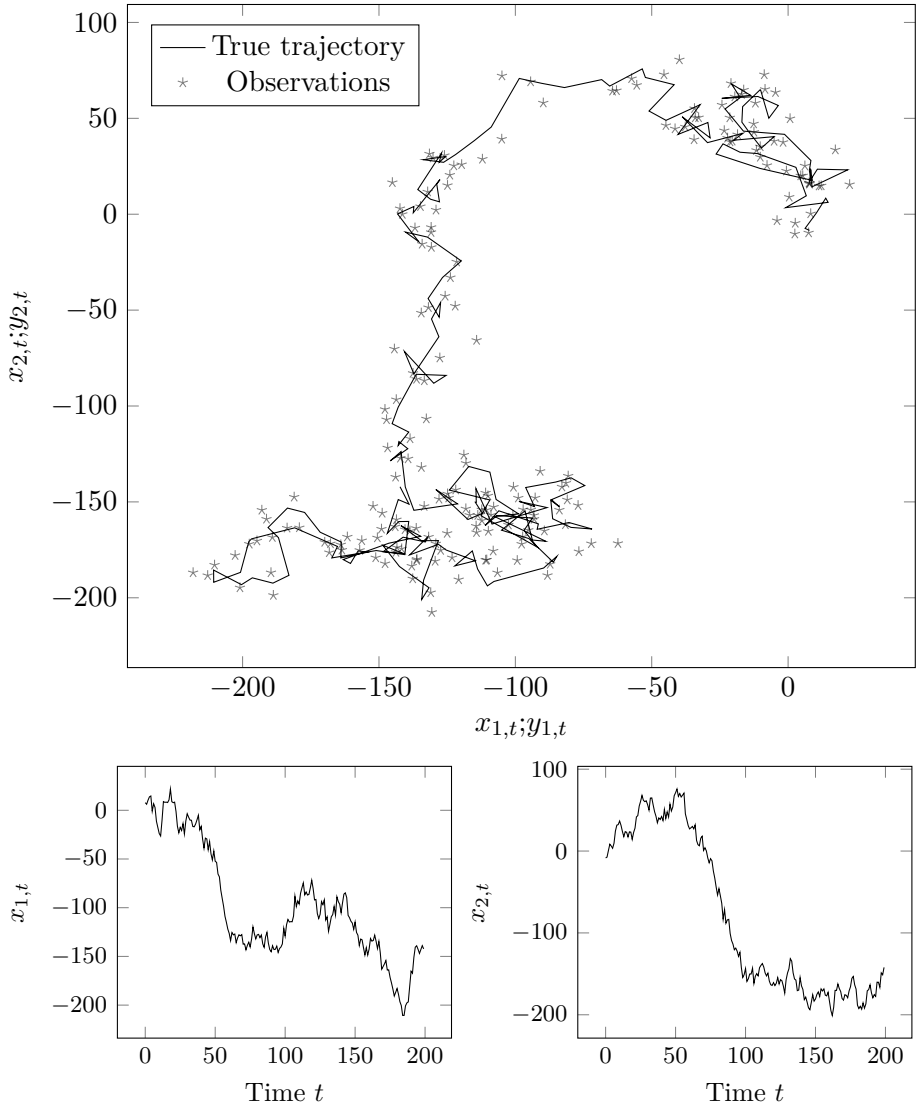

Figure 1.1: An example of a RWM simulated trajectory

with covariance matrix

$$Q = \begin{bmatrix} \frac{\Delta_t^3}{3} & 0 & \frac{\Delta_t^2}{2} & 0 \\ 0 & \frac{\Delta_t^3}{3} & 0 & \frac{\Delta_t^2}{2} \\ \frac{\Delta_t^2}{2} & 0 & \Delta_t & 0 \\ 0 & \frac{\Delta_t^2}{2} & 0 & \Delta_t \end{bmatrix} \widetilde{q},$$

where Δ_t is sampling period and $\widetilde{q}$ is process noise intensity. The observation matrix in the observation equation (1.4) is

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The observation noise v_t is an independent, zero-mean variable

$$v_t \sim \mathcal{WN}(0, R_t), \quad R_t \in \mathbb{R}^{2\times 2}.$$

1.3.3 Constant acceleration model

Similarly to the previous model, this model assumes constant acceleration between successive measurements, and changes in acceleration (also known as "jerk") are modeled as a zero-mean white noise. In addition to the target location and velocity, the acceleration components $a_{x_1,t}$ and $a_{x_2,t}$ are included in the state vector. The equations for target dynamics are as follows

$$\begin{aligned} x_{1,t} &= x_{1,t-1} + v_{x_1,t}\Delta_t + \frac{1}{2}a_{x_1,t}\Delta_t^2 + w_{x_1,t}, \\ x_{2,t} &= x_{2,t-1} + v_{x_2,t}\Delta_t + \frac{1}{2}a_{x_2,t}\Delta_t^2 + w_{x_2,t}, \\ v_{x_1,t} &= v_{x_1,t-1} + a_{x_1,t}\Delta_t + w_{v_{x_1},t}, \\ v_{x_2,t} &= v_{x_2,t-1} + a_{x_2,t}\Delta_t + w_{v_{x_2},t}, \\ a_{x_1,t} &= a_{x_1,t-1} + w_{a_{x_1},t}, \\ a_{x_2,t} &= a_{x_2,t-1} + w_{a_{x_2},t}. \end{aligned} \tag{1.7}$$

By rewriting these equations into matrix form (1.3), we get the following state vector

$$x_t = \begin{bmatrix} x_{1,t} \\ x_{2,t} \\ v_{x_1,t} \\ v_{x_2,t} \\ a_{x_1,t} \\ a_{x_2,t} \end{bmatrix}$$

and state transition matrix

$$A = \begin{bmatrix} 1 & 0 & \Delta_t & 0 & \frac{1}{2}\Delta_t^2 & 0 \\ 0 & 1 & 0 & \Delta_t & 0 & \frac{1}{2}\Delta_t^2 \\ 0 & 0 & 1 & 0 & \Delta_t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta_t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The independent process noise w_t is given by

$$w_t = \begin{bmatrix} w_{x_1,t} \\ w_{x_2,t} \\ w_{v_{x_1},t} \\ w_{v_{x_2},t} \\ w_{a_{x_1},t} \\ w_{a_{x_2},t} \end{bmatrix}, \quad w_t \sim \mathcal{WN}(0, Q),$$

with covariance matrix

$$Q = \begin{bmatrix} \frac{\Delta_t^5}{20} & 0 & \frac{\Delta_t^4}{8} & 0 & \frac{\Delta_t^3}{6} & 0 \\ 0 & \frac{\Delta_t^5}{20} & 0 & \frac{\Delta_t^4}{8} & 0 & \frac{\Delta_t^3}{6} \\ \frac{\Delta_t^4}{8} & 0 & \frac{\Delta_t^3}{3} & 0 & \frac{\Delta_t^2}{2} & 0 \\ 0 & \frac{\Delta_t^4}{8} & 0 & \frac{\Delta_t^3}{3} & 0 & \frac{\Delta_t^2}{2} \\ \frac{\Delta_t^3}{6} & 0 & \frac{\Delta_t^2}{2} & 0 & \Delta_t & 0 \\ 0 & \frac{\Delta_t^3}{6} & 0 & \frac{\Delta_t^2}{2} & 0 & \Delta_t \end{bmatrix} \widetilde{q},$$

where Δ_t is sampling period and $\widetilde{q}$ is process noise intensity. The observation equation is of the usual form (1.4) and

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The observation noise v_t is an independent, zero-mean variable

$$v_t \sim \mathcal{WN}(0, R_t), \quad R_t \in \mathbb{R}^{2\times 2}.$$

1.4 Bayesian sequential estimation of discrete-time state-space models

In Bayesian estimation of state-space models, the model (1.1) is reformulated into conditional probability distribution form [3] with probability density functions (pdfs)

$$x_t | x_{t-1}, u_t \sim \pi(x_t | x_{t-1}, u_t), \tag{1.8}$$

$$y_t | x_t \sim p(y_t | x_t). \tag{1.9}$$

The former (1.8) is usually called the state evolution or the state transition model, the latter (1.9) is the measurement model or the observation model.

The problem of state estimation is to evaluate $\pi(x_t|Y_t, U_t)$, the distribution of x_t given observations $Y_t = \{y_1, ..., y_t\}$ and inputs $U_t = \{u_1, ..., u_t\}$ [7, Chap. 6]. Let the prior pdf of x_{t-1} be $\pi(x_{t-1}|Y_{t-1}, U_{t-1})$, the Bayesian sequential (online) estimation runs in two steps:

1. *Prediction* [8] - the predicted state x_t from x_{t-1} is obtained using the prior pdf $\pi(x_{t-1}|Y_{t-1}, U_{t-1})$, the state evolution model (1.8) and the Chapman-Kolmogorov equation as

$$\pi(x_t|U_t, Y_{t-1}) = \int \pi(x_t|x_{t-1}, u_t)\pi(x_{t-1}|Y_{t-1}, U_{t-1})dx_{t-1}. \tag{1.10}$$

2. *Update* - the distribution $\pi(x_t|U_t, Y_{t-1})$, now the prior, is updated with the information about x_t from the collected observation y_t via the Bayes' theorem:

$$\pi(x_t|Y_t, U_t) = \frac{\pi(x_t|U_t, Y_{t-1})p(y_t|x_t)}{\int \pi(x_t|U_t, Y_{t-1})p(y_t|x_t)dx_t}. \tag{1.11}$$

The problem of the update step is that the posterior pdf is usually not analytically tractable and has to be approximated, e. g., using Markov chain Monte Carlo (MCMC) methods, which is not suitable for online sequential estimation. However, if the observation model $p(y_t|x_t)$ belongs to the exponential family of distributions and the prior $\pi(x_t|U_t, Y_{t-1})$ is conjugate to it, a closed form posterior distribution exists [8, 9, 10].

Definition 1 (Exponential family of distributions)**.** Let y_t be a random variable parametrized by vector θ_t. The distribution of y_t belongs to the exponential family if its pdf can be written in the form

$$f(y_t|\theta_t) = h(y_t) \exp\left(\eta^\intercal T - A(\eta)\right), \tag{1.12}$$

where $\eta \equiv \eta(\theta_t)$ is the natural parameter, $T \equiv T(y_t)$ is a sufficient statistic which contains all the information y_t provides about the parameter θ_t. $h(y_t)$ is a known function independent of θ_t, and $A(\eta)$ is a known log-normalizing function.

Definition 2 (Conjugate prior)**.** Consider a random variable y_t obeying an exponential family distribution parametrized by θ_t. The prior distribution of θ_t characterized by the hyper-parameters ξ_{t-1} of the same dimension as the sufficient statistic T and $\nu_{t-1} \in \mathbb{R}^+$ is said to be conjugate to it, if its pdf can be written in the form

$$\pi(\theta_t|\xi_{t-1}, \nu_{t-1}) = q(\xi_{t-1}, \nu_{t-1}) \exp\left(\eta^\intercal \xi_{t-1} - \nu_{t-1}A(\eta)\right), \tag{1.13}$$

where $q(\xi_{t-1}, \nu_{t-1})$ is the normalizing function. The terms $A(\eta)$ and η are the same as in the exponential family.

Many well-known distributions, such as Bernoulli, multinomial, Poisson, and gamma, are of the exponential family. For this thesis, the normal distribution will be of particular interest. The topic of conjugate priors is thoroughly discussed in [9].

Under conjugacy, the Bayesian update (1.11), which involves multiplying the model $p(y_t|x_t)$ with the prior $\pi(x_t|U_t, Y_{t-1})$, yields the posterior $\pi(x_t|Y_t, U_t)$ of the same type as the prior, its pdf has the hyper-parameters

$$\begin{aligned} \xi_t &= \xi_{t-1} + T(y_t), \\ \nu_t &= \nu_{t-1} + 1, \end{aligned} \tag{1.14}$$

where $T(y_t)$ is the sufficient statistic of the model $p(y_t|x_t)$, and ξ_{t-1} and ν_{t-1} are the hyper-parameters of the conjugate prior $\pi(x_t|U_t, Y_{t-1})$. Thus, the Bayesian update is reduced to a simple update of the conjugate prior's hyper-parameters.

Chapter 2

Distributed estimation

In recent decades, estimation in decentralized distributed networks has gained significant research interest due to the increasing availability of low-cost sensors, their versatility, and a diverse range of applications including environmental monitoring, spacecraft navigation, airborne target tracking, applications in medicine, etc. [11, 12, 13]. Each node in the distributed architecture communicates only with its directly connected neighbors. Compared to centralized architectures with one or more information processing nodes (fusion centers), this approach could decrease the communication burden and provide improved robustness against node failure. Depending on how nodes communicate with their neighbors, the distributed information processing schemes can be classified into three types of strategies: incremental, consensus-based, and diffusion.

The incremental strategy requires agents[1] to be connected in a Hamiltonian cycle. This strategy, however, provides limited robustness to node and link failures as each node and link represent a single point of failure. Additionally, recovery from a failure requires finding a new Hamiltonian cyclic path which is generally an NP-hard problem. Furthermore, the cooperation between nodes is limited, each node is only allowed to receive information from one predecessor and share information with one successor [14, 8].

The objective of the consensus-based methods is two-fold: (i) constructing a local estimate based on its own observations and estimates from its neighbors and (ii) reach a consensus with neighboring estimators on the state estimate, e.g., an average value [15, 16]. Communication burden in consensus-based strategies can be lowered by implementing a gossip protocol, i.e., at a random time instant, an agent selects a random neighbor and averages or swaps their local states [17, 18].

In the diffusion strategies, networks are represented by directed or undirected connected graphs with node degrees usually higher than one. Commu-

[1] in this text, the terms node and agent are used interchangeably

nication occurs only locally between adjacent nodes, and no consensus on the estimate is expected. Conventionally, the diffusion strategy at each time step runs in two phases - (i) an adaptation phase, the agent obtains its neighbors' measurements and incorporates them into its local estimate and (ii) a combination phase that merges the neighbors' estimates [8]. The belief is that over time the knowledge of each agent is gradually diffused through the whole network. This thesis adopts Bayesian formulation of the adapt-then-combine (ATC) diffusion strategy presented in [8].

2.1 Estimation by diffusion

Let us consider a network characterized by a connected undirected graph with a set of nodes $\mathcal{I} = \{1, 2, ..., I\}$. Each node $i \in \mathcal{I}$ is only allowed to communicate with its adjacent neighbors forming its neighborhood $\mathcal{I}^{(i)}$ with cardinality $|\mathcal{I}^{(i)}|$, also assume that $i \in \mathcal{I}^{(i)}$. In diffusion networks, there are two types of information exchange: the exchange of observations is called the *adaptation* phase and the exchange of estimates takes place during the *combination* phase. Following up on Section 1.4, these phases will be described below in terms of Bayesian probability theory. Since it is possible to utilize either one or both phases, we discriminate between four different schemes: adaptation only (A), combination only (C), adapt-then-combine (ATC), and combine-then-adapt (CTA). In the sequel, we will mostly focus on the ATC strategy because it has been shown experimentally in [14] and theoretically in [8] that it outperforms the CTA scheme.

Remark. The combination-only scheme can be useful, e.g., when the communication cost of the measurement exchange significantly outweighs the communication cost of the estimate exchange.

2.1.1 Adaptation phase

The goal of this phase is to enrich the statistical knowledge of every node by incorporating its neighbors' observations. Fixing a node $i \in \mathcal{I}$, node i acquires observations $y_t^{(j)}$ from its neighbors $j \in \mathcal{I}^{(i)}$. Each observation is independently assigned an adaptation weight $c_{ij,t} \in [0, 1]$, this expresses the degree of node i's belief in jth node's information. Alternatively, this can be interpreted as the subjective probability that jth node's information is true (from node i's perspective) [3]. In the context of Bayesian theory, this means to perform a weighted Bayesian batch update (or $|\mathcal{I}^{(i)}|$ weighted Bayesian sequential updates)

$$\pi^{(i)}(x_t|\tilde{Y}_t^{(i)}, \tilde{U}_t^{(i)}) \propto \pi^{(i)}(x_t|\tilde{U}_t^{(i)}, \tilde{Y}_{t-1}^{(i)}) \prod_{j \in \mathcal{I}^{(i)}} p(y_t^{(j)}|x_t)^{c_{ij,t}}. \tag{2.1}$$

The $\propto$ is the proportionality operator, it means equality up to the normalization factor. The tilde notation indicates the variables affected by the shared information. Similarly to (1.14), assuming $\xi_{t-1}^{(i)}$ and $\nu_{t-1}^{(i)}$ to be the hyper-parameters of the prior $\pi^{(i)}(x_t|\tilde{U}_t^{(i)}, \tilde{Y}_{t-1}^{(i)})$, then under conjugacy, the hyper-parameters of the posterior $\pi^{(i)}(x_t|\tilde{Y}_t^{(i)}, \tilde{U}_t^{(i)})$ are

$$\begin{aligned} \xi_t^{(i)} &= \xi_{t-1}^{(i)} + \sum_{j \in \mathcal{I}^{(i)}} c_{ij,t} T(y_t^{(j)}), \\ \nu_t^{(i)} &= \nu_{t-1}^{(i)} + \sum_{j \in \mathcal{I}^{(i)}} c_{ij,t}. \end{aligned} \tag{2.2}$$

2.1.2 Combination phase

In the ATC scheme, the combination phase now proceeds with the posterior pdfs produced in the adaptation phase. The objective of the combination phase is to share and collaboratively improve the estimates of the agents by merging of the posterior pdfs from the adaptation step [8]. In Bayesian theory, an established measure of discrepancy (divergence) between pdfs is the Kullback-Leibler divergence (KLD) [19]

$$\mathcal{D}_{KL}(p||q) = \mathbb{E}_{p(\theta)}\left[\log \frac{p(\theta)}{q(\theta)}\right] = \int_\theta p(\theta) \log \frac{p(\theta)}{q(\theta)} d\theta, \tag{2.3}$$

$p(\theta)$ and $q(\theta)$ are probability density functions.

The goal is to find a final density $\tilde{\pi}^{(i)}(\cdot) \equiv \tilde{\pi}^{(i)}(x_t|\tilde{Y}_t^{(i)}, \tilde{U}_t^{(i)})$, such that the divergence from the densities $\pi^{(j)}(\cdot) \equiv \pi^{(j)}(x_t|\tilde{Y}_t^{(j)}, \tilde{U}_t^{(j)})$ of all its neighbors $j \in \mathcal{I}^{(i)}$ is minimal. In other words, we wish to minimize the loss

$$\sum_{j \in \mathcal{I}^{(i)}} a_{ij,t} \mathcal{D}_{KL}\left(\tilde{\pi}^{(i)}(\cdot) \middle|\middle| \pi^{(j)}(\cdot)\right), \tag{2.4}$$

where $a_{ij,t} \in [0, 1]$ summing to unity are combination weights, expressing node i's degree of belief in node j's information.

The solution and proof of optimality in the KLD sense is given in [8, Prop. 1] and [10, Prop. 1]; its form is the following:

$$\tilde{\pi}^{(i)}(x_t|\tilde{Y}_t^{(i)}, \tilde{U}_t^{(i)}) \propto \prod_{j \in \mathcal{I}^{(i)}} \pi^{(j)}(x_t|\tilde{Y}_t^{(j)}, \tilde{U}_t^{(j)})^{a_{ij,t}}. \tag{2.5}$$

The combination of exponential family densities again leads to an analytically tractable solution. Assuming conjugate posterior pdfs from the adaptation phase $\pi^{(j)}(\cdot)$ with hyper-parameters $\xi_t^{(j)}$ and $\nu_t^{(j)}$. The combination step

yields a posterior pdf $\tilde{\pi}^{(i)}(\cdot)$ with the hyper-parameters

$$\begin{aligned} \tilde{\xi}_t^{(i)} &= \sum_{j \in \mathcal{I}^{(i)}} a_{ij,t} \xi_t^{(j)}, \\ \tilde{\nu}_t^{(i)} &= \sum_{j \in \mathcal{I}^{(i)}} a_{ij,t} \nu_t^{(j)}. \end{aligned} \tag{2.6}$$

In the event the ATC strategy is employed, the result of the combination phase is the final estimation product of the diffusion step at time instant t. Moreover, it can serve as the prior for the next adaptation phase at $t+1$ [8].

Chapter 3

Diffusion Kalman filter

3.1 Kalman filter

Consider a linear state-space model (1.2) from Section 1.2 driven by mutually independent normal zero-mean noise variables

$$w_t \sim \mathcal{N}(0, Q_t),$$
$$v_t \sim \mathcal{N}(0, R_t).$$

Using the Bayesian approach from Section 1.4, we formulate the model in terms of conditional probability distributions

$$x_t|x_{t-1}, u_t \sim \mathcal{N}(A_t x_{t-1} + B_t u_t, Q_t), \tag{3.1}$$
$$y_t|x_t \sim \mathcal{N}(H_t x_t, R_t), \tag{3.2}$$

where $x_t \in \mathbb{R}^n$ are state vectors, $u_t \in \mathbb{R}^n$ are known inputs, $y_t \in \mathbb{R}^k$ are observations. A_t, B_t, and H_t are known matrices of compatible dimensions and $Q_t \in \mathbb{R}^{n\times n}$, $R_t \in \mathbb{R}^{k\times k}$ are covariance matrices. The pdfs of (3.1), (3.2) are $\pi(x_t|x_{t-1}, u_t)$ and $p(y_t|x_t)$, respectively.

As demonstrated in Section 1.4, the filtration runs in two steps: prediction and update.

3.1.1 Prediction

The prediction step applies the state evolution model (3.1) to the estimate of x_{t-1} according to the Equation (1.10), that is

$$\pi(x_t|U_t, Y_{t-1}) = \int \pi(x_t|x_{t-1}, u_t)\pi(x_{t-1}|Y_{t-1}, U_{t-1})dx_{t-1}.$$

The estimate of x_{t-1} is represented by a normal prior pdf $\pi(x_{t-1}|Y_{t-1}, U_{t-1})$ with mean $\hat{x}^+_{t-1}$ and covariance matrix P^+_{t-1} (at $t = 1$ we assume initial state

$x_0 \sim \mathcal{N}(\hat{x}_0^+, P_0^+)$). Note that the state evolution model (3.1) is also a normal pdf. As a consequence, the integrand is a product of two normal pdfs, this results in joint multivariate normal pdf of both x_t and x_{t-1}. We can then use Lemma 2 to get the solution of the integral. As a result of Lemma 2, the nuisance parameters not relevant to x_t are marginalized out. Furthermore, the result is again a normal pdf $\pi(x_t|U_t, Y_{t-1})$ with mean $\hat{x}_t^-$ and covariance matrix P_t^- given by

$$\begin{aligned} \hat{x}_t^- &= A_t \hat{x}_{t-1}^+ + B_t u_t, \\ P_t^- &= A_t P_{t-1}^+ A_t^\intercal + Q_t. \end{aligned}$$

3.1.2 Update

The update step incorporates a new measurement y_t by means of the Bayes' theorem as described in (1.11).

We notice that the observation model $y_t|x_t \sim \mathcal{N}(H_t x_t, R_t)$. The normal distribution belongs to the exponential family (see Definition 1) and its pdf $p(y_t|x_t)$ can be written in the form

$$p(y_t|x_t) = (2\pi)^{-\frac{k}{2}} (\det R_t)^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2}(y_t - H_t x_t)^\intercal R_t^{-1}(y_t - H_t x_t) \right\} \tag{3.3}$$

$$\propto \exp\left\{ -\frac{1}{2}\mathrm{Tr}\left(\underbrace{\begin{bmatrix} -1 \\ x_t \end{bmatrix} \begin{bmatrix} -1 \\ x_t \end{bmatrix}^\intercal}_{\eta} \underbrace{\begin{bmatrix} y_t^\intercal \\ H_t^\intercal \end{bmatrix} R_t^{-1} \begin{bmatrix} y_t^\intercal \\ H_t^\intercal \end{bmatrix}^\intercal}_{T(y_t)} \right) \right\}, \tag{3.4}$$

where (3.4) corresponds to the exponential family form seen in Definition 1. A simplified notation with the trace operator is used to avoid vectorization of the involved matrices. The estimate from the prediction step — now the prior — also happens to be a normal distribution with mean $\hat{x}_t^-$ and covariance matrix P_t^-. Its pdf written in the compatible conjugate form (1.13) reads as follows

$$\pi(x_t|U_t, Y_{t-1}) \propto \exp\left\{ -\frac{1}{2}\mathrm{Tr}\left(\underbrace{\begin{bmatrix} -1 \\ x_t \end{bmatrix} \begin{bmatrix} -1 \\ x_t \end{bmatrix}^\intercal}_{\eta} \underbrace{\begin{bmatrix} (\hat{x}_t^-)^\intercal \\ I \end{bmatrix} (P_t^-)^{-1} \begin{bmatrix} (\hat{x}_t^-)^\intercal \\ I \end{bmatrix}^\intercal}_{\xi_{t-1}} \right) \right\}. \tag{3.5}$$

To get a better understanding of the whole process, let us take a closer look at the matrices ξ_{t-1} and $T(y_t)$.

$$\xi_{t-1} = \begin{bmatrix} (\hat{x}_t^-)^\intercal \\ I \end{bmatrix} (P_t^-)^{-1} \begin{bmatrix} (\hat{x}_t^-)^\intercal \\ I \end{bmatrix}^\intercal \tag{3.6}$$

$$= \begin{bmatrix} (\hat{x}_t^-)^\intercal (P_t^-)^{-1} \hat{x}_t^- & (\hat{x}_t^-)^\intercal (P_t^-)^{-1} \\ (P_t^-)^{-1} \hat{x}_t^- & (P_t^-)^{-1} \end{bmatrix}, \tag{3.7}$$

$$T(y_t) = \begin{bmatrix} y_t^\intercal \\ H_t^\intercal \end{bmatrix} R_t^{-1} \begin{bmatrix} y_t^\intercal \\ H_t^\intercal \end{bmatrix}^\intercal \tag{3.8}$$

$$= \begin{bmatrix} y_t^\intercal R_t^{-1} y_t & y_t^\intercal R_t^{-1} H_t \\ H_t^\intercal R_t^{-1} y_t & H_t^\intercal R_t^{-1} H_t \end{bmatrix} \tag{3.9}$$

We can now perform the hyper-parameter update according to the Formula (1.14)[2] as

$$\xi_t = \xi_{t-1} + T(y_t) \tag{3.10}$$

$$= \begin{bmatrix} (\hat{x}_t^-)^\intercal (P_t^-)^{-1} \hat{x}_t^- + y_t^\intercal R_t^{-1} y_t & (\hat{x}_t^-)^\intercal (P_t^-)^{-1} + y_t^\intercal R_t^{-1} H_t \\ \underbrace{(P_t^-)^{-1} \hat{x}_t^- + H_t^\intercal R_t^{-1} y_t}_{(P_t^+)^{-1} \hat{x}_t^+} & \underbrace{(P_t^-)^{-1} + H_t^\intercal R_t^{-1} H_t}_{(P_t^+)^{-1}} \end{bmatrix} \tag{3.11}$$

Recall that under conjugacy, the posterior distribution is of the same type as the prior distribution. As a result, the posterior is also a normal distribution with mean $\hat{x}_t^+$ and covariance matrix P_t^+, we denote its pdf $\pi(x_t|Y_t, U_t)$. We can exploit the blocks of the matrix ξ_t (3.11) to retrieve the original parameters of its pdf in this way

$$\begin{aligned} P_t^+ &= (\xi_{t;[2,2]})^{-1} \\ &= \left[(P_t^-)^{-1} + H_t^\intercal R_t^{-1} H_t\right]^{-1} \\ &= P_t^- - \underbrace{P_t^- H_t^\intercal (R_t + H_t P_t^- H_t^\intercal)^{-1}}_{K_t} H_t P_t^- \\ &= (I - K_t H_t) P_t^- \end{aligned} \tag{3.12}$$

and

$$\begin{aligned} \hat{x}_t^+ &= \overbrace{(\xi_{t;[2,2]})^{-1}}^{P_t^+} \overbrace{(\xi_{t;[2,1]})}^{(P_t^+)^{-1}\hat{x}_t^+} \\ &= P_t^+ \left[(P_t^-)^{-1} \hat{x}_t^- + H_t^\intercal R_t^{-1} y_t\right] \\ &\vdots \\ &= \hat{x}_t^- + P_t^+ H_t^\intercal R_t^{-1} (y_t - H_t \hat{x}_t^-). \end{aligned} \tag{3.13}$$

[2]for Kalman filtering, the hyper-parameter ν is of no interest

To obtain (3.12) we used the the matrix inversion lemma[3] and K_t is called the Kalman gain, it is

$$K_t = P_t^- H_t^\intercal (R_t + H_t P_t^- H_t^\intercal)^{-1}.$$

In the literature, the update of the mean $\hat{x}_t^+$ is usually expressed in terms of the Kalman gain K_t. The formulation in (3.13) is mathematically equivalent [20]and will be beneficial later in the derivation of the diffusion Kalman filter.

The above derivation of the update step follows the the approach of K. Dedecius in [3, 8]. Reformulating the problem as an update of the conjugate prior's hyper-parameters yields arguably an algebraically easier approach to its derivation than traditional Bayesian approaches (see [7, 21, 22]). There are a number of mathematically equivalent forms of the Kalman filter, some of them are given in [20]. Besides that, there are, of course, extensions to the traditional Kalman filter, where the state evolution and observation models do not need to be linear functions, e.g., the extended Kalman filter (EKF) and the unscented Kalman filter (UKF), but they are beyond the scope of this thesis.

3.2 Diffusion Kalman filter

With the "traditional" Kalman filter now in place, we can apply the theory from Section 2.1 to derive the diffusion version of the filter. Suppose a diffusion network with the node set $\mathcal{I}$, focusing on a fixed node $i \in \mathcal{I}$ with neighborhood $\mathcal{I}^{(i)}$ (i belongs in $\mathcal{I}^{(i)}$ too), the estimation proceeds in three stages: *local prediction*, *adaptation*, and *combination*. In the local prediction stage, node i performs a local next state prediction as described in Section 3.1.1:

$$\pi^{(i)}(x_t|\tilde{U}_t^{(i)}, \tilde{Y}_{t-1}^{(i)}) = \int \pi(x_t|x_{t-1}, u_t) \underbrace{\pi^{(i)}(x_{t-1}|\tilde{Y}_{t-1}^{(i)}, \tilde{U}_{t-1}^{(i)})}_{\text{prior}} dx_{t-1},$$

where the variables $\tilde{Y}_{t-1}^{(i)}$ and $\tilde{U}_{t-1}^{(i)}$ of the prior represent all the information available to node i at the time instant $t-1$. The result of the local prediction step is a normal distribution with the transformed mean and covariance matrix

$$\begin{aligned} \hat{x}_t^{(i),-} &= A_t \tilde{x}_{t-1}^{(i),+} + B_t u_t, \\ P_t^{(i),-} &= A_t \tilde{P}_{t-1}^{(i),+} A_t^\intercal + Q_t. \end{aligned} \tag{3.14}$$

3.2.1 Adaptation phase

During the adaptation step, nodes update their estimates by their local and neighbors' measurements by virtue of the Bayes' theorem from (2.1). The

[3] $(A+UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$

adaptation step is derived by utilizing the theory developed in Sections 2.1.1 and 3.1.2, it is the distributed counterpart to the local Kalman filter update.

Node i receives observations $y_t^{(j)}$ from its neighbors $j \in \mathcal{I}^{(i)}$ (recall that $i \in \mathcal{I}^{(i)}$ too). Noting that both the prior and the observation model are again normally distributed and are thus conjugate, the Formulae (2.1) and (2.2) can be directly implemented to get the posterior distribution. Similarly to (3.4) and (3.5), the observation model and the prior can be written in the exponential family (1.12) and the conjugate prior (1.13) forms, respectively.

$$p(y_t^{(j)}|x_t) \propto \exp\left\{ -\frac{1}{2}\mathrm{Tr}\Bigg(\underbrace{\begin{bmatrix}-1\\ x_t\end{bmatrix}\begin{bmatrix}-1\\ x_t\end{bmatrix}^\intercal}_{\eta} \underbrace{\begin{bmatrix}(y_t^{(j)})^\intercal \\ \left(H_t^{(j)}\right)^\intercal\end{bmatrix}\left(R_t^{(j)}\right)^{-1}\begin{bmatrix}(y_t^{(j)})^\intercal \\ \left(H_t^{(j)}\right)^\intercal\end{bmatrix}^\intercal}_{T\left(y_t^{(j)}\right)} \Bigg)\right\} \tag{3.15}$$

$$\begin{aligned}\pi^{(i)}(x_t|\tilde{U}_t^{(i)}, \tilde{Y}_{t-1}^{(i)}) \\ \propto \exp\left\{ -\frac{1}{2}\mathrm{Tr}\Bigg(\underbrace{\begin{bmatrix}-1\\ x_t\end{bmatrix}\begin{bmatrix}-1\\ x_t\end{bmatrix}^\intercal}_{\eta} \underbrace{\begin{bmatrix}(\hat{x}_t^{(i),-})^\intercal \\ I\end{bmatrix}\left(P_t^{(i),-}\right)^{-1}\begin{bmatrix}(\hat{x}_t^{(i),-})^\intercal \\ I\end{bmatrix}^\intercal}_{\xi_{t-1}^{(i)}} \Bigg)\right\}\end{aligned} \tag{3.16}$$

The problem is again reduced to a simple update of the conjugate prior's hyper-parameter $\xi_{t-1}^{(i)}$ by the sufficient statistics $T\left(y_t^{(j)}\right)$. Updating the prior's hyper-parameters according to (2.2), we get the matrix

$$\begin{aligned}\xi_t^{(i)} &= \xi_{t-1}^{(i)} + \sum_{j\in\mathcal{I}^{(i)}} c_{ij,t} T\left(y_t^{(j)}\right) \\ &= \begin{bmatrix}(\hat{x}_t^{(i),-})^\intercal \\ I\end{bmatrix}\left(P_t^{(i),-}\right)^{-1}\begin{bmatrix}(\hat{x}_t^{(i),-})^\intercal \\ I\end{bmatrix}^\intercal \\ &+ \sum_{j\in\mathcal{I}^{(i)}} c_{ij,t}\begin{bmatrix}(y_t^{(j)})^\intercal \\ \left(H_t^{(j)}\right)^\intercal\end{bmatrix}\left(R_t^{(j)}\right)^{-1}\begin{bmatrix}(y_t^{(j)})^\intercal \\ \left(H_t^{(j)}\right)^\intercal\end{bmatrix}^\intercal .\end{aligned} \tag{3.17}$$

In case the original normal distribution parameters are wanted, they can be obtained from the blocks of the matrix $\xi_t^{(i)}$ (similar to Equations (3.12) and (3.13)), the covariance matrix takes the form

$$P_t^{(i),+} = \left(\xi_{t;[2,2]}^{(i)}\right)^{-1} = \left[P_t^{(i),-} + \sum_{j\in\mathcal{I}^{(i)}} c_{ij,t}\left(H_t^{(j)}\right)^\intercal\left(R_t^{(j)}\right)^{-1}H_t^{(j)}\right]^{-1} \tag{3.18}$$

and the mean

$$\begin{aligned}\hat{x}_t^{(i),+} &= P_t^{(i),+}\left(\xi_{t;[1,2]}^{(i)}\right)^{-1} = P_t^{(i),+}\left(P_t^{(i),+}\right)^{(-1)}\hat{x}_t^{(i),-} \\ &\vdots \\ &= \hat{x}_t^{(i),-} + P_t^{(i),+}\left[\sum_{j\in\mathcal{I}^{(i)}} c_{ij,t}\left(H_t^{(j)}\right)^{\intercal}\left(R_t^{(j)}\right)^{-1}\left(y_t^{(j)} - H_t^{(j)}\hat{x}_t^{(i),-}\right)\right].\end{aligned} \tag{3.19}$$

3.2.2 Combination phase

The aim of the combination stage is to improve the estimation performance by the exchange and combination of estimates within local neighborhoods. In the case of the diffusion Kalman filter, the estimates are represented by the normal posterior distributions obtained in the adaptation stage (Section 2.1.1) with pdfs $\pi^{(j)}(x_t|\tilde{Y}_t^{(j)}, \tilde{U}_t^{(j)})$ with means $\hat{x}_t^{(j),+}$ and covariance matrices $P_t^{(j),+}$, see Equations (3.19) and (3.18), respectively.

According to (2.5), the optimal way (in the KLD sense) to get the combination posterior $\tilde{\pi}^{(i)}(x_t|\tilde{Y}_t^{(i)}, \tilde{U}_t^{(i)})$ of these densities is the weighted geometric product

$$\begin{aligned}\tilde{\pi}^{(i)}(x_t|\tilde{Y}_t^{(i)}, \tilde{U}_t^{(i)}) &\propto \prod_{j\in\mathcal{I}^{(i)}} \pi^{(j)}(x_t|\tilde{Y}_t^{(j)}, \tilde{U}_t^{(j)})^{a_{ij,t}} \\ &= \prod_{j\in\mathcal{I}^{(i)}} [\mathcal{N}(\hat{x}_t^{(j),+}, P_t^{(j),+})]^{a_{ij,t}},\end{aligned} \tag{3.20}$$

which minimizes the cumulative KLD loss (2.4).

Since the normal distribution belongs to the exponential family (Definition 1), then by applying (2.6), the product of normal densities is reduced to a weighted sum of the matrices $\xi_t^{(j)}$ from the adaptation stage (3.17). As a result, the matrix $\tilde{\xi}_t^{(i)}$ of the combination posterior takes the form

$$\begin{aligned}\tilde{\xi}_t^{(i)} &= \sum_{j\in\mathcal{I}^{(i)}} a_{ij,t}\xi_t^{(j)} \\ &= \sum_{j\in\mathcal{I}^{(i)}} a_{ij,t}\begin{bmatrix}(\hat{x}_t^{(j),+})^{\intercal}\left(P_t^{(j),+}\right)^{-1}\hat{x}_t^{(j),+} & (\hat{x}_t^{(j),+})^{\intercal}\left(P_t^{(j),+}\right)^{-1} \\ \left(P_t^{(j),+}\right)^{-1}\hat{x}_t^{(j),+} & \left(P_t^{(j),+}\right)^{-1}\end{bmatrix} \\ &= \begin{bmatrix}(\tilde{x}_t^{(i),+})^{\intercal}\left(\tilde{P}_t^{(i),+}\right)^{-1}\tilde{x}_t^{(i),+} & (\tilde{x}_t^{(i),+})^{\intercal}\left(\tilde{P}_t^{(i),+}\right)^{-1} \\ \left(\tilde{P}_t^{(i),+}\right)^{-1}\tilde{x}_t^{(i),+} & \left(\tilde{P}_t^{(i),+}\right)^{-1}\end{bmatrix}.\end{aligned} \tag{3.21}$$

Similarly to Equations (3.18) and (3.19), the matrix $\tilde{\xi}_t^{(i)}$ contains the original normal distribution parameters, a little algebra reveals

$$\begin{aligned} \tilde{P}_t^{(i),+} &= \Big[\sum_{j \in \mathcal{I}^{(i)}} a_{ij,t} \Big(P_t^{(j),+} \Big)^{-1} \Big]^{-1}, \\ \tilde{x}_t^{(i),+} &= \tilde{P}_t^{(i),+} \Big(\sum_{j \in \mathcal{I}^{(i)}} a_{ij,t} \Big(P_t^{(j),+} \Big)^{-1} \hat{x}_t^{(j),+} \Big). \end{aligned} \tag{3.22}$$

This result is also known as the *covariance intersection* [23].

Remark. The original diffusion Kalman filter (diffKF), first suggested by the authors Cattivelli and Sayed in [1], has the same adaptation phase as the algorithm described above, but chooses to only combine the point estimates in the combination phase, leaving the associated covariance matrices intact. This filter was later extended by Hu et al. in [2] by incorporating a procedure involving covariance intersection, yielding an algorithm equivalent to the one described above (and in [8]).

Chapter 4

Distributed Kalman filtering under partially heterogeneous models

Assume a diffusion network with nodes $i \in \mathcal{I}$. At each discrete time instant $t = 1, 2, ...,$ network nodes acquire uni- or multivariate noisy observations $y_t^{(i)} \in \mathbb{R}^k$ of a hidden Markov process, nodes may employ different models of various complexity in their estimation

$$\begin{aligned} x_t^{(i)} &= A_t^{(i)} x_{t-1}^{(i)} + B_t^{(i)} u_t^{(i)} + w_t^{(i)}, \\ y_t^{(i)} &= H_t^{(i)} x_t^{(i)} + v_t^{(i)}, \end{aligned} \tag{4.1}$$

where $x_t^{(i)} \in \mathbb{R}^{n^{(i)}}$ are state variables and $u_t^{(i)}$ (if they exists) are known input variables. While observations $y_t^{(i)}$ are different for each node, the state-space of the observations is assumed to be homogeneous. Matrices $A_t^{(i)}, B_t^{(i)}$ and $H_t^{(i)}$ are known and of compatible dimensions. $w_t^{(i)}$ are the independent process noise variables

$$w_t^{(i)} \sim \mathcal{N}(0, Q_t^{(i)}), \quad Q_t^{(i)} \in \mathbb{R}^{n^{(i)} \times n^{(i)}}.$$

The observation noise variables $v_t^{(i)}$ are also zero-mean, independent and identically distributed with known covariance matrices $R_t^{(i)}$

$$v_t^{(i)} \sim \mathcal{N}(0, R_t^{(i)}), \quad R_t^{(i)} \in \mathbb{R}^{k \times k}.$$

The state-space model from 4.1 can take the form of probability density functions

$$\pi_i(x_t^{(i)} | x_{t-1}^{(i)}, u_t^{(i)}) \equiv \mathcal{N}(A_t^{(i)} x_{t-1}^{(i)} + B_t^{(i)} u_t^{(i)}, Q_t^{(i)}), \tag{4.2}$$

$$p_i(y_t^{(i)} | x_t^{(i)}) \equiv \mathcal{N}(H_t^{(i)} x_t^{(i)}, R_t^{(i)}). \tag{4.3}$$

Although, in general, the nodes may utilize completely heterogeneous state-space models, in our example, we will limit ourselves to only partially heterogeneous state-space models. The nodes of the network employ state-space models $M_1, ..., M_K$, let $S_1, ..., S_K$ be their respective state-spaces, where $1 \leq K \leq |\mathcal{I}|$. The models are ordered by their increasing level of complexity, $dim(S_k) \leq dim(S_l)$, furthermore, $S_k \preccurlyeq S_l$ for all $k \leq l$, where $k, l \in K$. The $\preccurlyeq$ symbol indicates that S_k is a subspace of S_l. In other words, the state-space of the less complex (or underspecified) models is a subspace of more complex model state-space. For example, it is easy to see that for the kinematic models mentioned in Section 1.3

$$S_{\text{RWM}} \preccurlyeq S_{\text{CVM}} \preccurlyeq S_{\text{CAM}}$$

applies. Naturally, in an isolated environment with no collaboration among the nodes, the estimation performance and stability of the more complex models is significantly better than that of the underspecified models.

Figures 4.1, 4.3 and 4.5 show the performance of the isolated linear estimators (Kalman filters) in terms of average root mean square error (RMSE) over 100 independent runs of increasingly complex simulated 2D trajectories. Figures 4.2, 4.4 and 4.6 show the state and state estimate evolution for the simulated trajectories. It can be seen, that the RWM estimator's performance drastically degrades with the addition of velocity to the simulated trajectories, the degradation is further amplified with the addition of acceleration. The CVM estimator handles the simulated CAM trajectories much better, still, its performance seems to be decreasing over time, albeit at a much lower rate than that of the RWM estimator. Finally, while the CAM estimator shows lower performance than the true simulation models in the absence of acceleration, its performance quickly stabilizes at an acceptable level and does not drastically degrade over time. This is an expected result, the goal of the sequel is to propose a method, which improves the estimation performance of the underspecified models.

The reason why RWM cannot keep up with a trajectory simulated by CAM lies in its the process noise matrix Q_{RWM}. The changes in position $x_{1,t}$ and $x_{2,t}$ between successive measurements of a CAM trajectory are caused by three things: (i) process noise, (ii) velocity, and (iii) acceleration. The RWM completely misses the information about (ii) and (iii), it instead tries to unsuccessfully attribute all the variations in position to noise. Thus, causing an early filter failure. On the other hand, the CAM estimator, whose process noise Q_{CAM} takes into account both the velocity and acceleration, has little to no problems with keeping up. The problem can be somewhat alleviated for the RWM, provided that the variances of both velocity and acceleration in the data generating model are low, by using a larger Q_{RWM}, i. e., scaling it by a "reasonable" constant. In the opposite case, the scaling would have to be done periodically, e. g., at every time instant t. The effect can be seen in Figure 4.7.

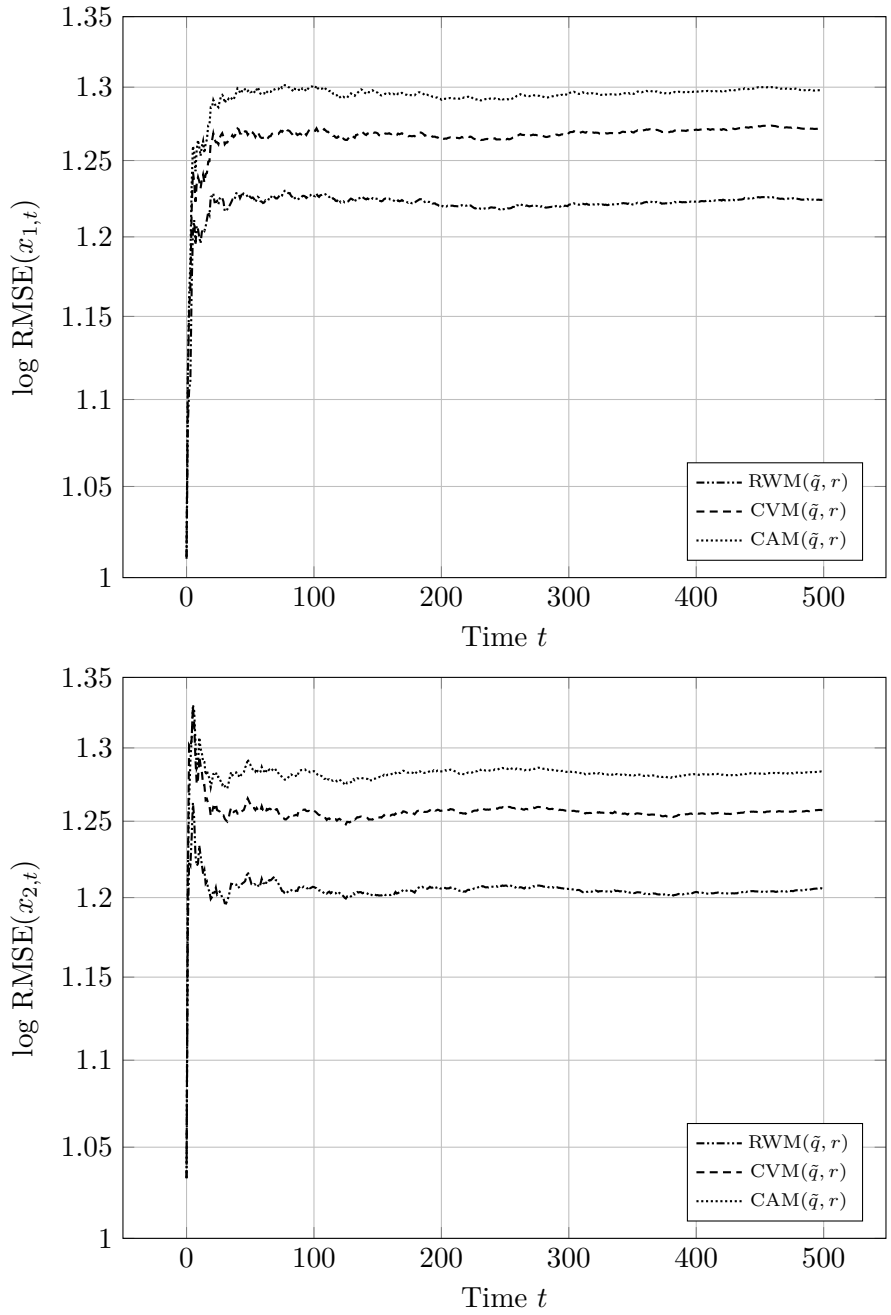

Figure 4.1: Decimal logarithm of the average RMSE over 100 independent runs of state estimates of a RWM trajectory with process noise intensity $\tilde{q} = 10$ and observation noise standard deviation $r = 1.3$.

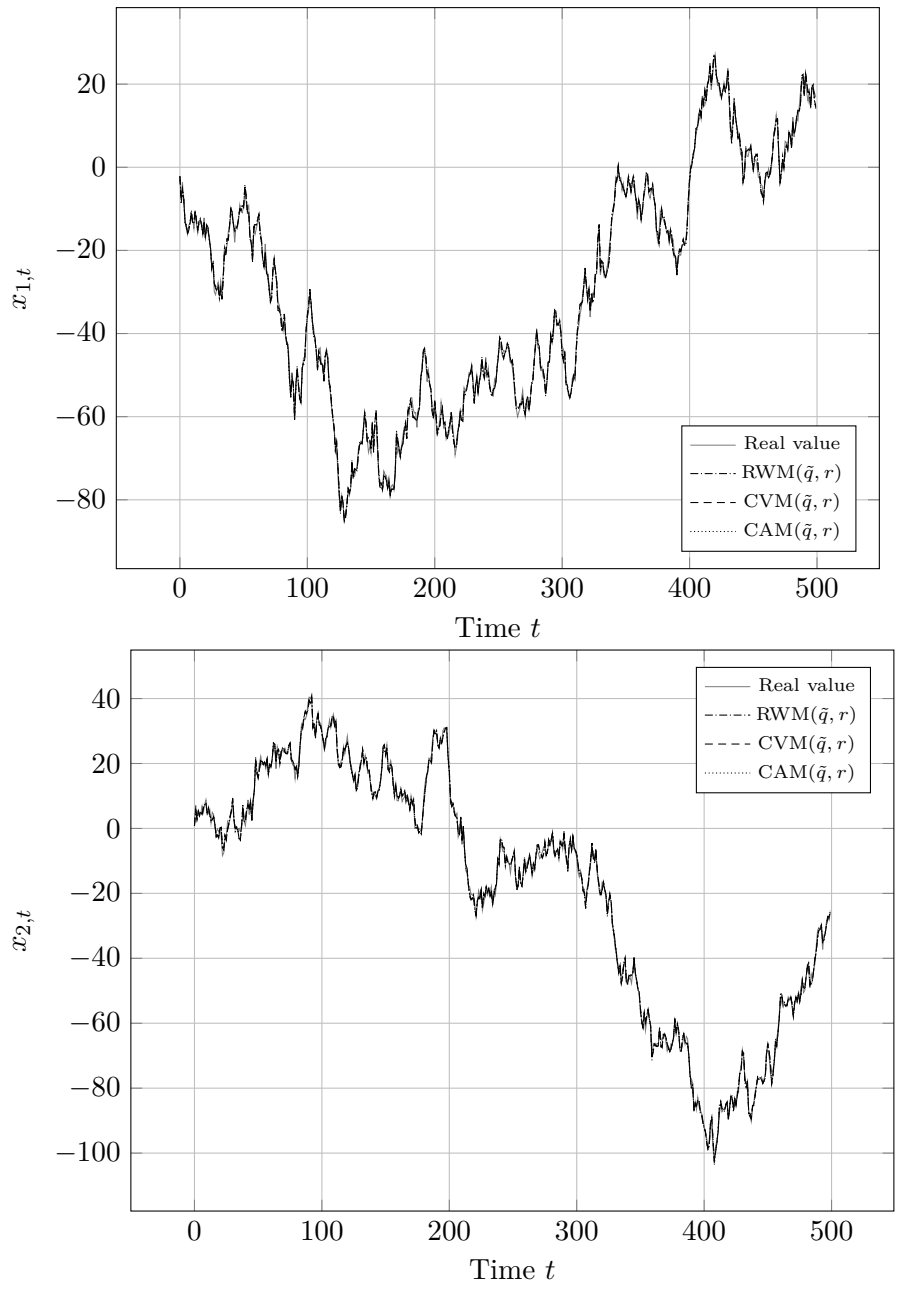

Figure 4.2: State and state estimate evolution of a randomly selected run for a RWM trajectory with process noise intensity $\tilde{q} = 10$ and observation noise standard deviation $r = 1.3$.

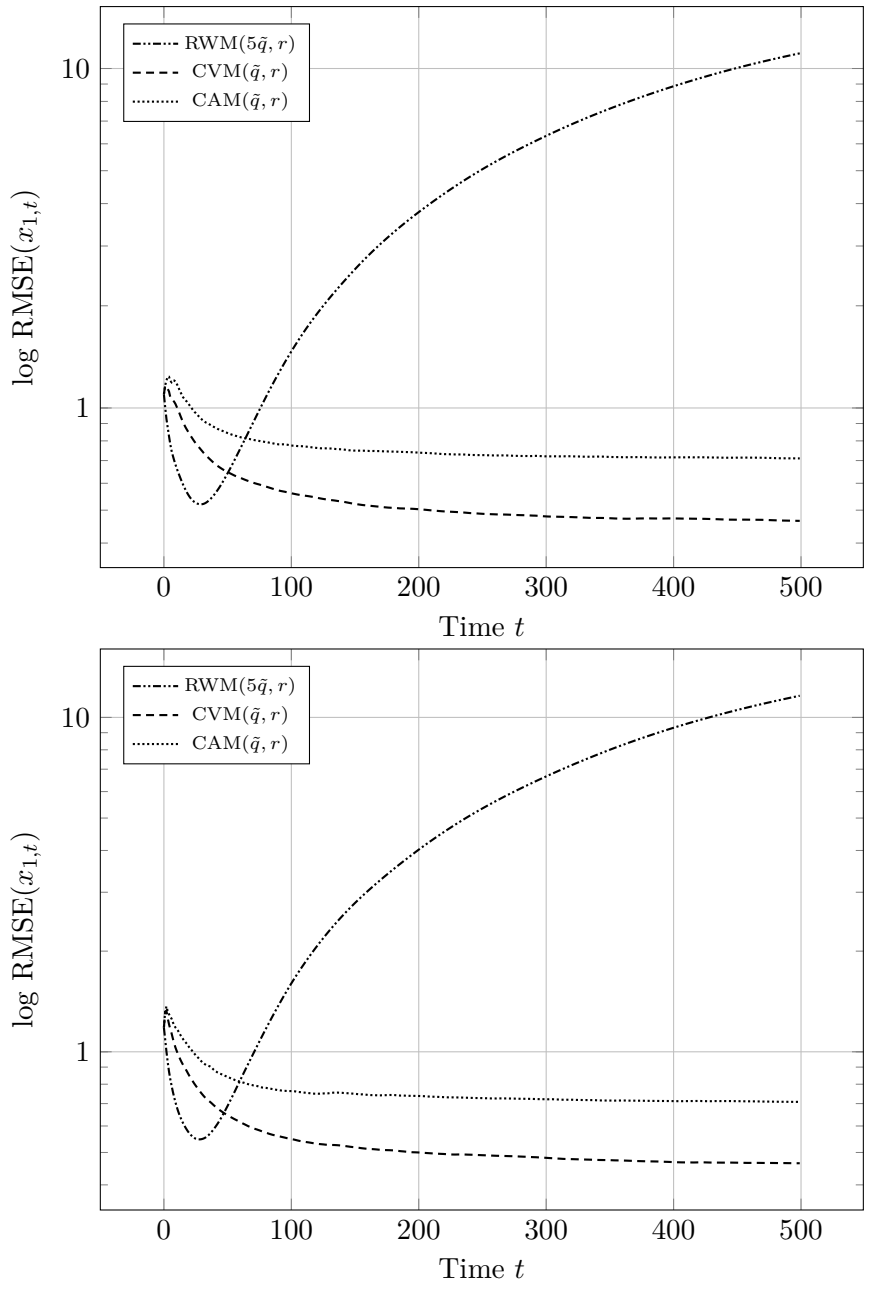

Figure 4.3: Decimal logarithm of the average RMSE over 100 independent runs of state estimates of a CVM trajectory with process noise intensity $\tilde{q} = 9.5 \times 10^{-5}$ and observation noise standard deviation $r = 1.3$.

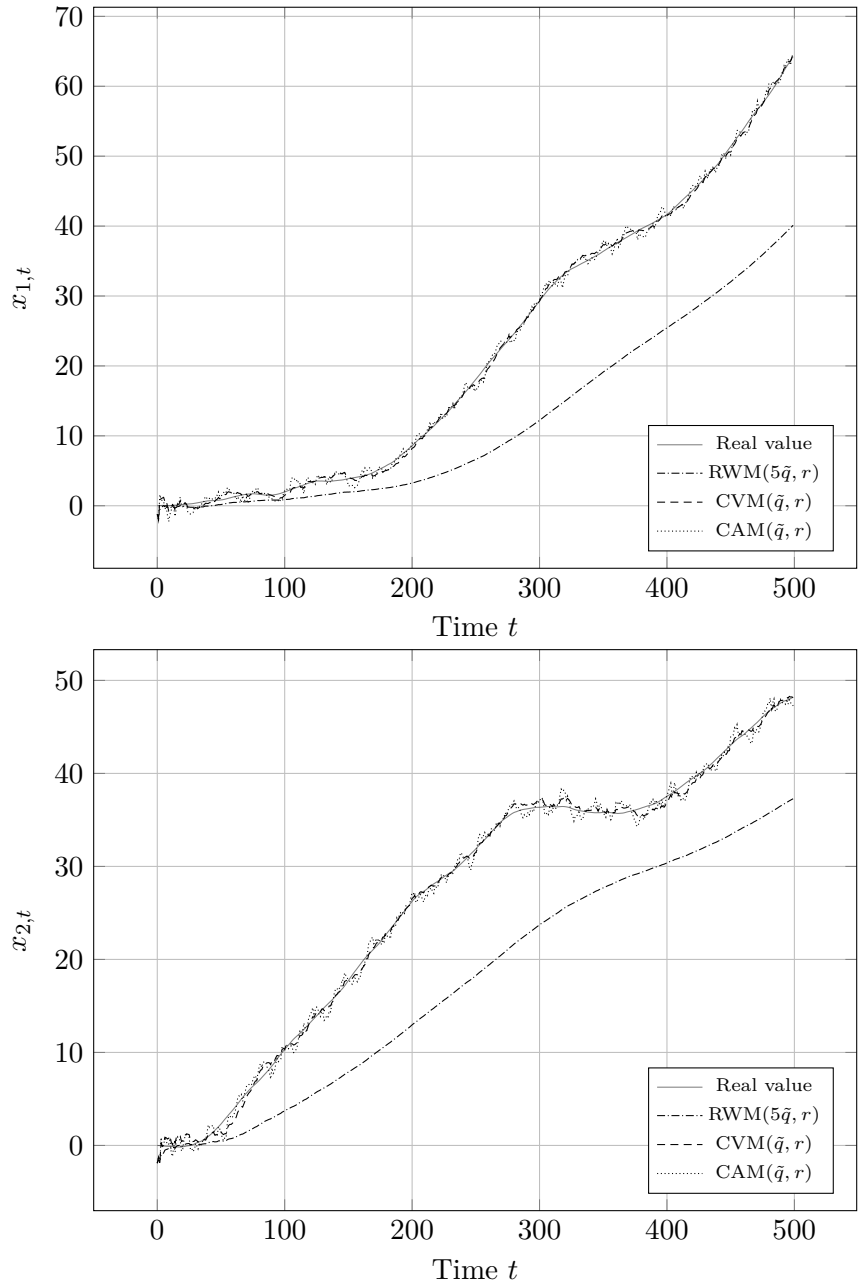

Figure 4.4: State and state estimate evolution of a randomly selected run for a CVM trajectory with process noise intensity $\tilde{q} = 9.5 \times 10^{-5}$ and observation noise standard deviation $r = 1.3$.

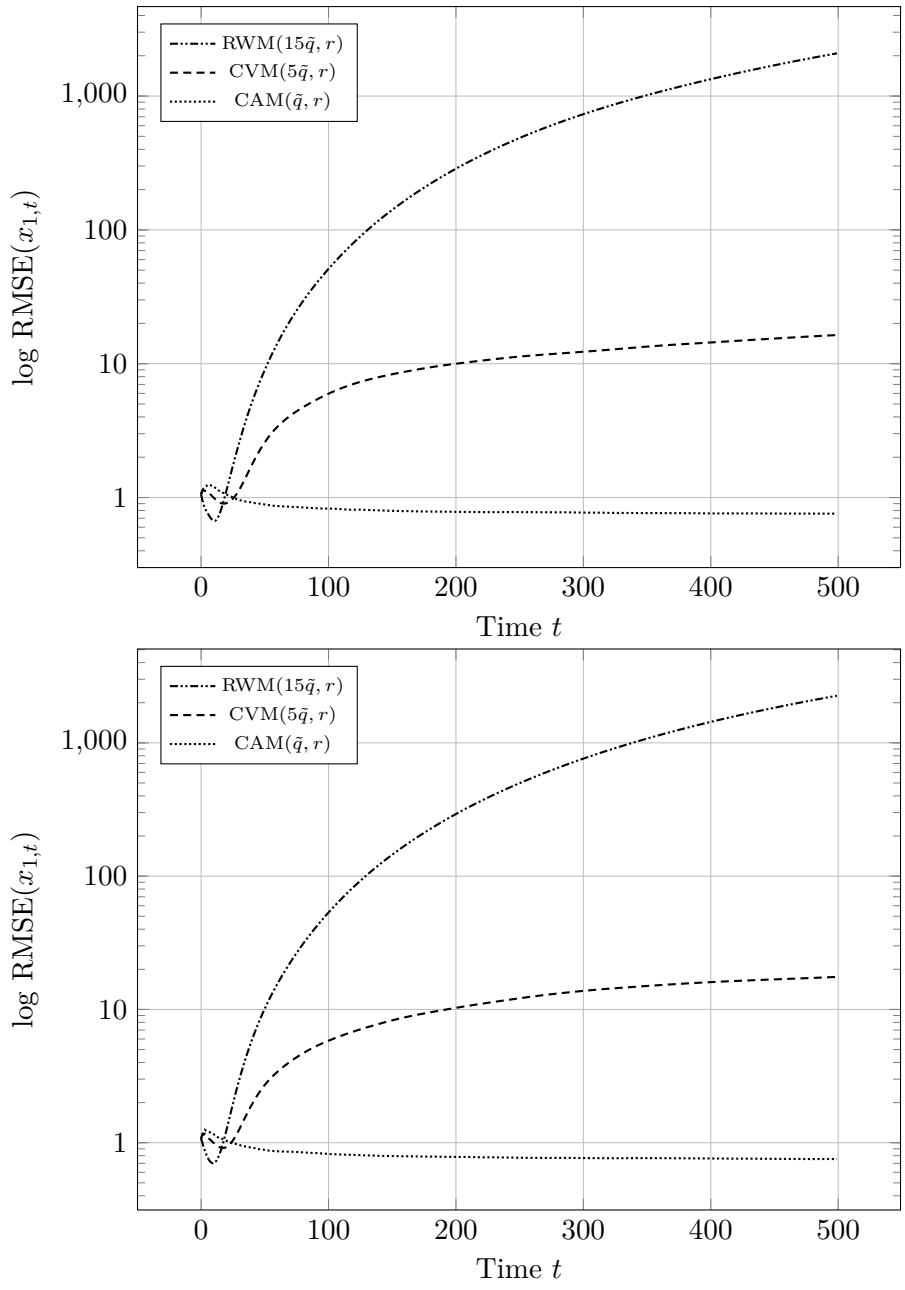

Figure 4.5: Decimal logarithm of the average RMSE over 100 independent runs of state estimates of a CAM trajectory with process noise intensity $\tilde{q} = 9.5 \times 10^{-5}$ and observation noise standard deviation $r = 1.3$.

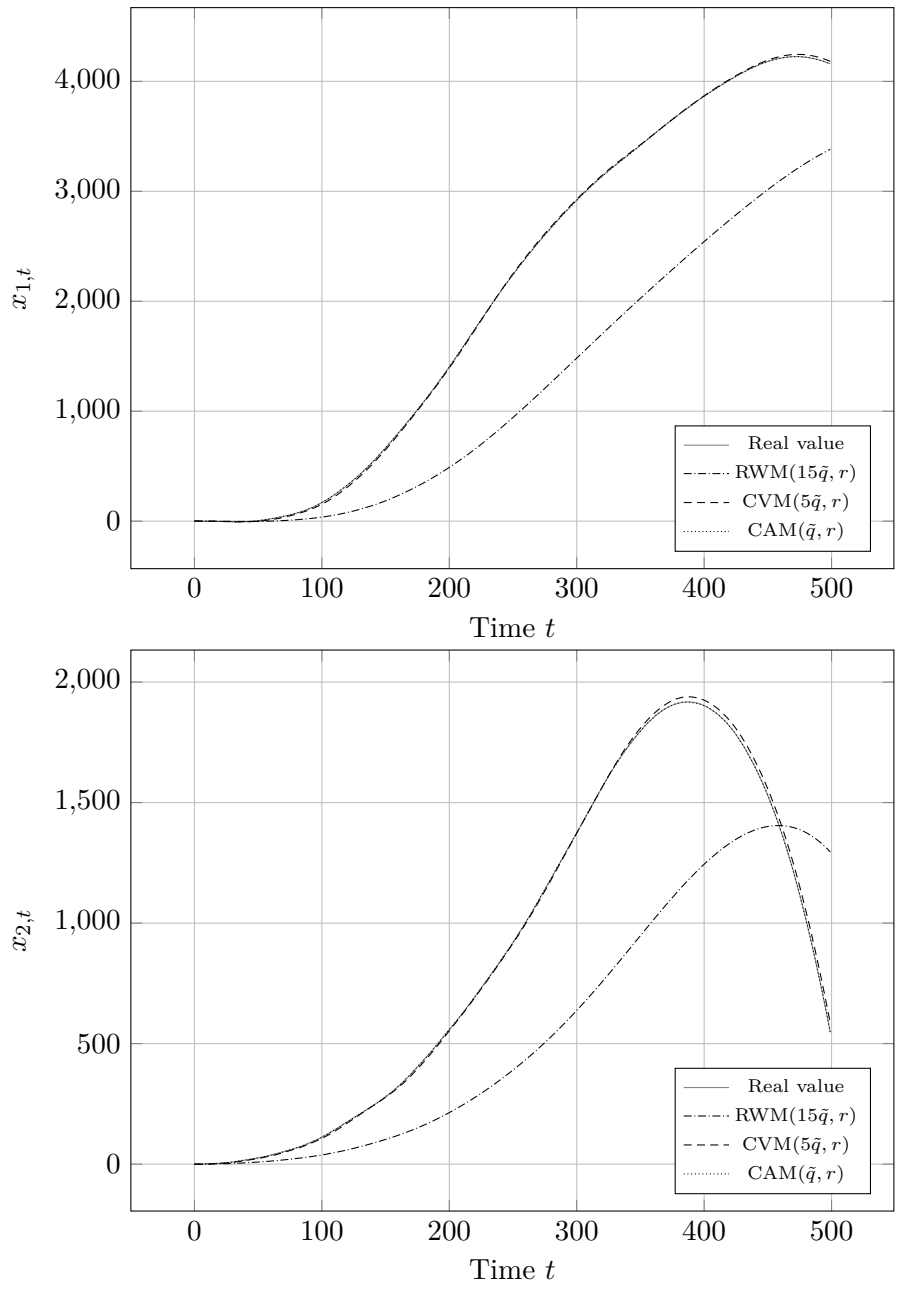

Figure 4.6: State and state estimate evolution of a randomly selected run for a CAM trajectory with process noise intensity $\tilde{q} = 9.5 \times 10^{-5}$ and observation noise standard deviation $r = 1.3$.

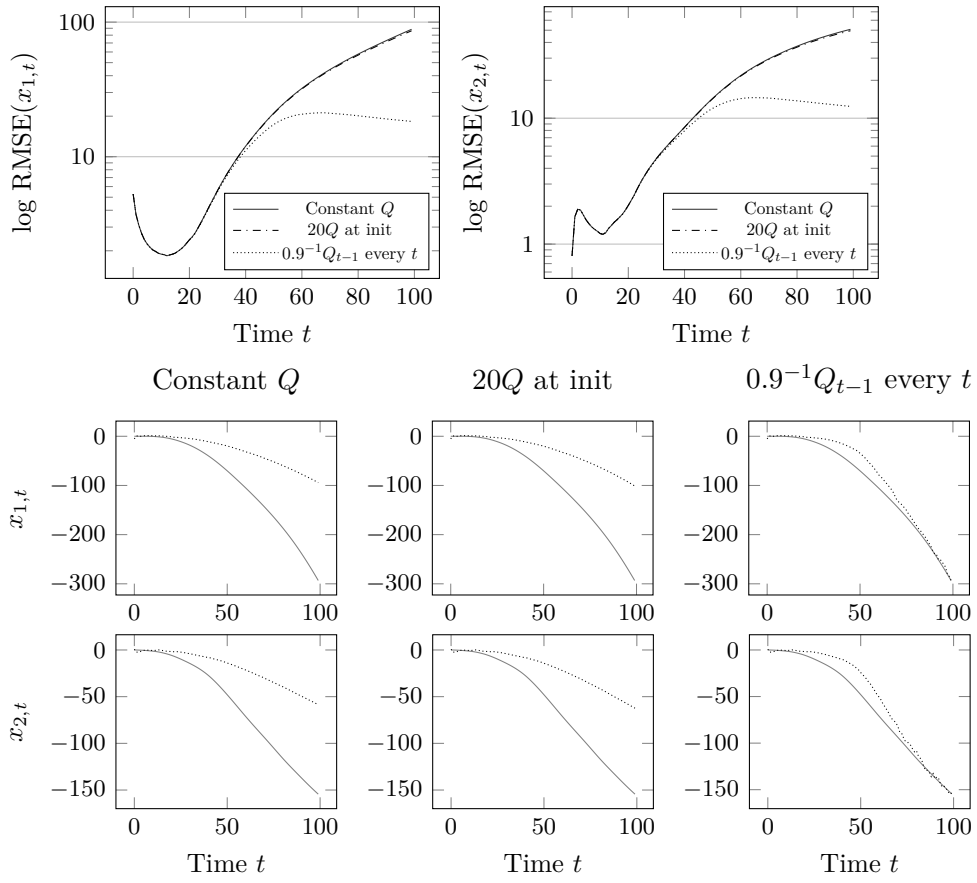

Figure 4.7: The effect of a larger covariance matrix Q_{RWM} on a CAM trajectory with process noise intensity $\hat{q} = 5 \cdot 10^{-4}$ and observation std $r = 5$. The dotted lines indicates the state estimate, the full file indicates the true value.

4.1 Exponential forgetting

Forgetting methods are often used in Bayesian estimation of time-varying parameters to stabilize the estimation process and avoid the growing discrepancy between the model and the true data generating process. The simplest, yet quite an effective technique is exponential forgetting [21, 24]. In the case of Kalman filtering, this means flattening of the prior distribution by its exponentiation before performing the prediction step (see Section 3.1.1)

$$\pi(x_{t-1}|Y_{t-1}, U_{t-1}) = \left[\pi(x_{t-1}|Y_{t-1}, U_{t-1})\right]^{\lambda} \tag{4.4}$$

$$= \left[\mathcal{N}(x^{+}_{t-1}, P^{+}_{t-1})\right]^{\lambda}, \quad \lambda \in [0, 1]. \tag{4.5}$$

Casting the prior in terms of the conjugate hyper-parameters ξ_{t-1} and ν_{t-1} (see Definition 2), this results in

$$\begin{aligned} \xi_{t-1} &= \lambda \xi_{t-1}, \\ \nu_{t-1} &= \lambda \nu_{t-1}, \end{aligned} \tag{4.6}$$

and

$$\xi_{t-1} = \lambda \begin{bmatrix} (x_{t-1}^{+})^{\intercal}(P_{t-1}^{+})^{-1}x_{t-1}^{+} & (x_{t-1}^{+})^{\intercal}(P_{t-1}^{+})^{-1} \\ (P_{t-1}^{+})^{-1}x_{t-1}^{+} & (P_{t-1}^{+})^{-1} \end{bmatrix}. \tag{4.7}$$

From which the conventional parameters of the normal distribution can be acquired as shown in (3.12) and (3.13). Consequently, the uncertainty associated with the state estimate is slightly increased in order to suppress possibly outdated or incompatible incorporated information. Typically, the forgetting factor $\lambda \in [0.95, 1]$ but because of the possible model misspecification, a lower value of λ might not be unreasonable.

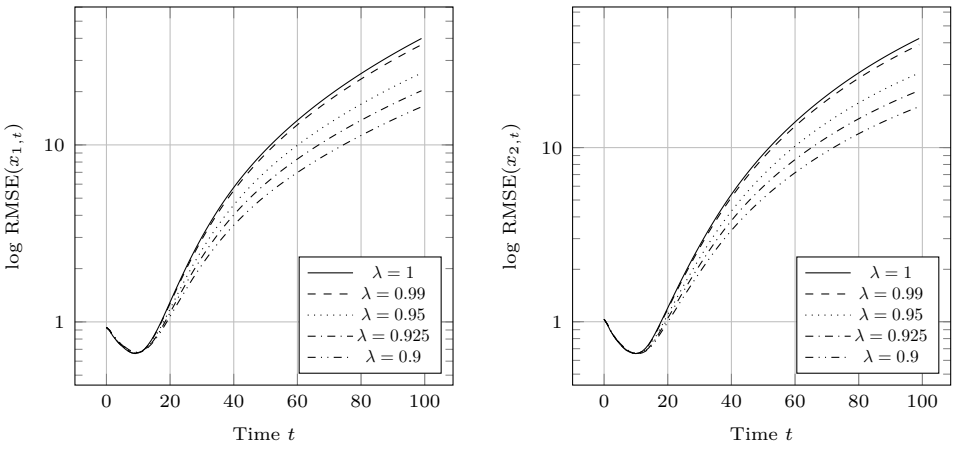

Figure 4.8: Evolution of the average RMSE for the RWM estimator with different values of λ estimating a CAM 2D trajectory.

Figures 4.8, 4.9 and 4.10, 4.11 show the effect of the forgetting factor λ on the misspecified models. The former show the evolution of average RMSE and the evolution of state estimates for the isolated RWM, the latter show the same for the isolated CVM. The average RMSE is calculated over 100 different independently simulated CAM trajectories, whereas the state estimate evolution is shown for one randomly selected run. The beneficial effect of the forgetting factor on the underspecified models can be clearly seen. Even though it cannot compensate for the model misspecification, exponential forgetting leads to a not insignificant improvement in estimation performance. On the contrary, in the case of the CAM estimator, the forgetting factor actually causes slight performance degradation, this is shown in Figure 4.12 (the

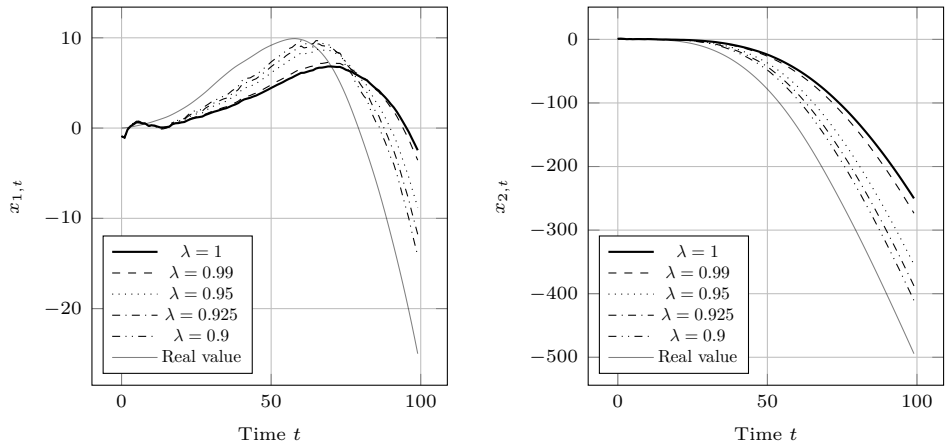

Figure 4.9: State and state estimate evolution of a randomly selected run of a RWM estimator for various values of λ estimating a 2D CAM trajectory.

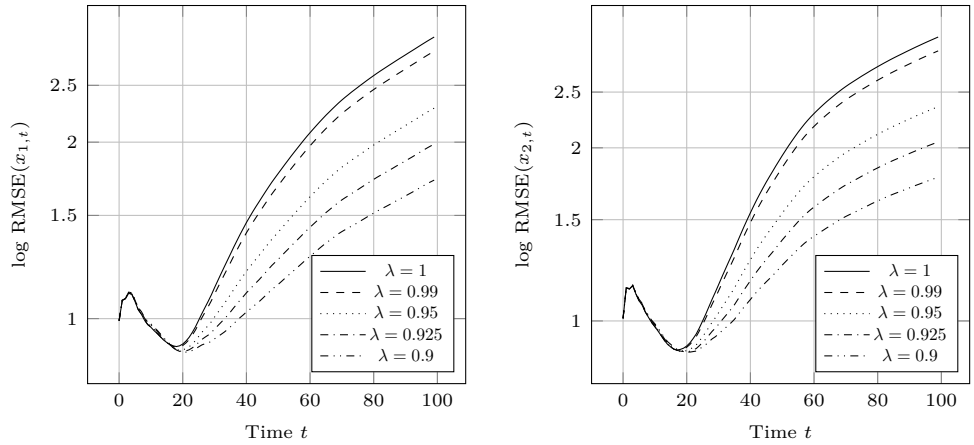

Figure 4.10: Evolution of the average RMSE for the CVM estimator with different values of λ estimating a CAM 2D trajectory.

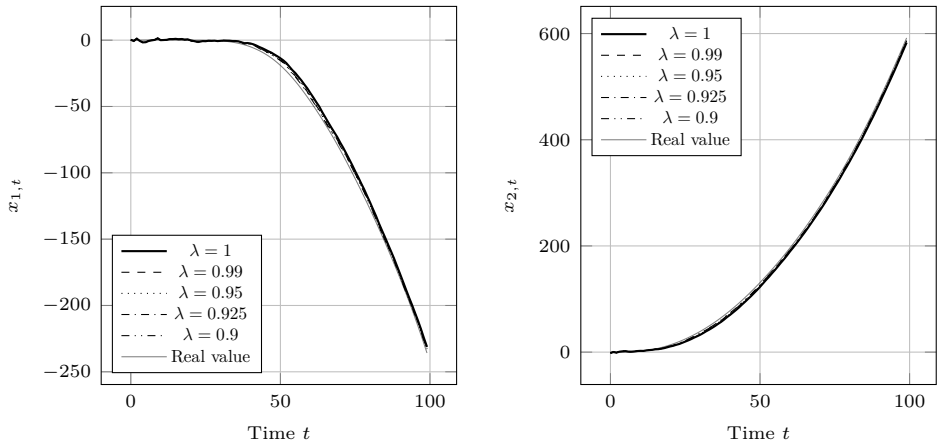

Figure 4.11: State and state estimate evolution of a randomly selected run of a CVM estimator for various values of λ estimating a 2D CAM trajectory.

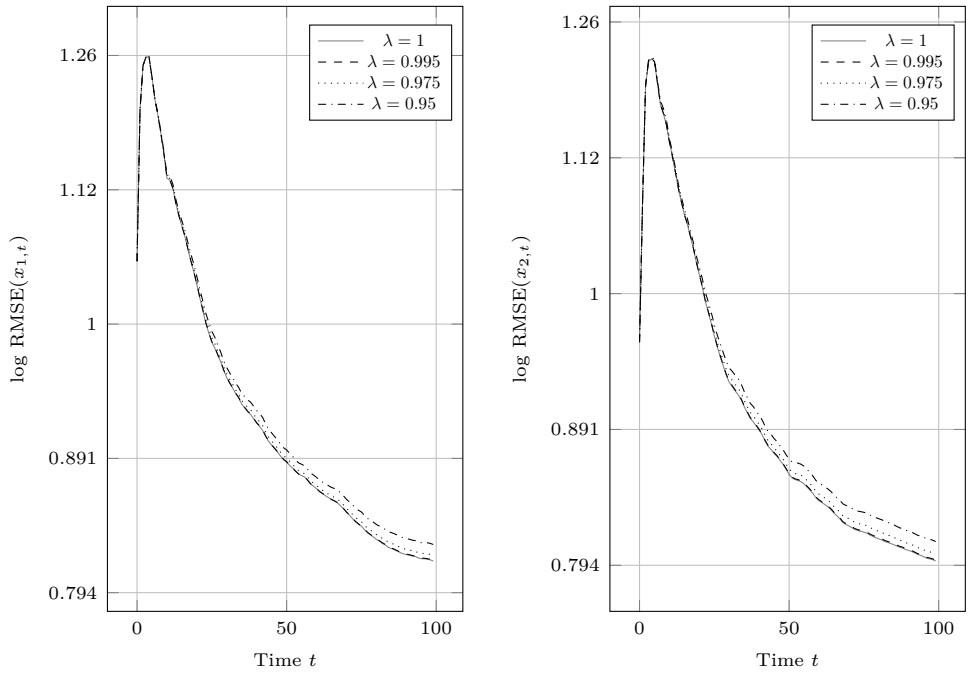

Figure 4.12: Evolution of the average RMSE for the CAM estimator with different values of λ estimating a CAM 2D trajectory.

state estimate evolution is omitted). Though this scenario might be rare in practice, as the estimation and simulation models are identical.

4.2 Estimation by diffusion

The diffusion Kalman filter in Section 3.2 assumes spatial homogeneity of the state-space models. However, with a few modifications, it can be adapted to the problem described at the beginning of this chapter. The main steps remain the same: *local prediction*, adaptation, and *combination*.

4.2.1 Local predication

The local prediction step remains mostly unchanged. The nodes $i \in \mathcal{I}$ locally perform the next state prediction according to their employed state evolution model (4.2)

$$\pi^{(i)}(x_t^{(i)}|\tilde{U}_t^{(i)}, \tilde{Y}_{t-1}^{(i)}) = \int \pi_i(x_t^{(i)}|x_{t-1}^{(i)}, u_t^{(i)})\pi^{(i)}(x_{t-1}^{(i)}|\tilde{Y}_{t-1}^{(i)}, \tilde{U}_{t-1}^{(i)})dx_{t-1}^{(i)},$$

where the variables $\tilde{Y}_{t-1}^{(i)}, \tilde{U}_{t-1}^{(i)}$ represent all the available information to the node i up to the time instant $t-1$ about the observation and input variables, respectively. This yields a normal distribution with the transformed mean and covariance matrix

$$\begin{aligned} \hat{x}_t^{(i),-} &= A_t^{(i)}\tilde{x}_{t-1}^{(i),+} + B_t^{(i)}u_t^{(i)}, \\ P_t^{(i),-} &= A_t^{(i)}\tilde{P}_{t-1}^{(i),+}\left(A_t^{(i)}\right)^{\mathsf{T}} + Q_t^{(i)}. \end{aligned} \tag{4.8}$$

4.2.2 Adaptation phase

Since we assume state-space homogeneity of the observation models (4.3), the adaptation stage from Section 2.1.1, during which nodes update their estimates by their local and neighbors' observations, can be straightforwardly applied to this diffusion estimator. The observations are incorporated by virtue of the Bayes' theorem

$$\pi^{(i)}(x_t^{(i)}|\tilde{Y}_t^{(i)}, \tilde{U}_t^{(i)}) \propto \pi^{(i)}(x_t^{(i)}|\tilde{U}_t^{(i)}, \tilde{Y}_{t-1}^{(i)}) \prod_{j\in\mathcal{I}^{(i)}} p_i(y_t^{(j)}|x_t^{(i)})^{c_{ij,t}}. \tag{4.9}$$

The resulting normal distribution has the following covariance matrix $P_t^{(i),+}$ and mean $\hat{x}_t^{(i),+}$

$$P_t^{(i),+} = \left[P_t^{(i),-} + \sum_{j\in\mathcal{I}^{(i)}} c_{ij,t}\left(H_t^{(i)}\right)^{\mathsf{T}}\left(R_t^{(j)}\right)^{-1}H_t^{(i)}\right]^{-1} \tag{4.10}$$

$$\hat{x}_t^{(i),+} = \hat{x}_t^{(i),-} + P_t^{(i),+}\left[\sum_{j\in\mathcal{I}^{(i)}} c_{ij,t}\left(H_t^{(i)}\right)^\intercal\left(R_t^{(j)}\right)^{-1}\left(y_t^{(j)} - H_t^{(i)}\hat{x}_t^{(i),-}\right)\right]. \tag{4.11}$$

4.2.3 Combination phase

In the combination phase from Section 3.2.2, the node i acquires the state estimates $\pi^{(j)}(\cdot)$ from its adjacent neighbors $j \in \mathcal{I}^{(i)}$ and combines them using the covariance intersection, creating a new state estimate $\tilde{\pi}^{(i)}(\cdot)$. The same philosophy will apply here, however, it would make sense to only incorporate estimates from models of the same or better complexity, as incorporating estimates of the underspecified models would most likely lead to degradation of estimation performance. Let us define the neighborhood $\mathcal{I}_S^{(i)} \subseteq \mathcal{I}^{(i)}$ containing only neighbors employing models of the same or better complexity than node i, i. e., $\mathcal{I}_S^{(i)} = \{k \in \mathcal{I}^{(i)} : S^{(i)} \preccurlyeq S^{(k)}\}$ (and i also belongs in $\mathcal{I}_S^{(i)}$).

The densities $\pi^{(j)}(\cdot)$ represent the neighbors' $j \in \mathcal{I}_S^{(i)}$ best estimate about the unknown hidden state $x_t^{(j)}$, but node i's interests lie in estimating the hidden state $x_t^{(i)}$, whose elements are a subset of $x_t^{(j)}$, possibly only a proper subset. The goal of node i is to obtain from $\pi^{(j)}(\cdot)$ only the probability distribution of variables contained within $x_t^{(i)}$, let us denote this distribution $\pi_{[i]}^{(j)}(\cdot)$. This is done by means of marginalization.

Lemma 1 (Marginal distribution of a continuous random vector [25])**.** *Consider a random vector* $\boldsymbol{X} = \left[X_1, \ldots, X_n\right]^\intercal$ *with joint probability density function* $f_{\boldsymbol{X}}(x_1, \ldots, x_n)$*. Then the marginal distribution of the random vector* $\boldsymbol{X}' = \left[X_{i_1}, \ldots, X_{i_k}\right]^\intercal$*, where* $k \leq n$ *and* $1 \leq i_1 < \ldots < i_k \leq n$*, has the probability density function*

$$f_{\boldsymbol{X}'}(x_{i_1}, \ldots, x_{i_k}) = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} f_{\boldsymbol{X}}(x_1, \ldots, x_n) dx_{j_1} \ldots dx_{j_{n-k}},$$

where $1 \leq j_1 < \ldots < j_{n-k} \leq n$ *and are different from* $i_1, \ldots, i_k$*.*

Lemma 1 says that we can get the distribution $\pi_{[i]}^{(j)}(\cdot)$ by integrating $\pi^{(j)}(\cdot)$ with respect to all variables except those in $x_t^{(i)}$,

$$\pi_{[i]}^{(j)}(x_t^{(j)}|\tilde{Y}_t^{(j)}, \tilde{U}_t^{(j)}) = \int \pi^{(j)}(x_t^{(j)}|\tilde{Y}_t^{(j)}, \tilde{U}_t^{(j)}) dx_{t,[j-i]}^{(j)}, \tag{4.12}$$

where $x_{t,[j-i]}^{(j)}$ denotes the state variables of node j of no interest to node i's model, i. e., the variables of node j's state estimate to be marginalized out. Thanks to the properties of the normal distribution, the integrand at

(4.12) can be straightforwardly evaluated via Lemma 2. It says that the undesirable elements of the mean vector $\hat{x}_t^{(j),+}$ and covariance matrix $P_t^{(j),+}$ can be simply discarded, this yields the parameters $\hat{x}_{t,[i]}^{(j),+}$ and $P_{t,[i]}^{(j),+}$ of the marginal normal distribution. The $[i]$ subscript notation indicates terms with variables irrelevant to node i marginalized out. An example follows.

Example. Consider a RWM estimate $\pi^{(i)}(x_t^{(i)}|\tilde{Y}_t^{(i)}, \tilde{U}_t^{(i)}) = \mathcal{N}(\hat{x}_t^{(i),+}, P_t^{(i),+})$ and a CVM estimate $\pi^{(j)}(x_t^{(j)}|\tilde{Y}_t^{(j)}, \tilde{U}_t^{(j)}) = \mathcal{N}(\hat{x}_t^{(j),+}, P_t^{(j),+})$ with

$$\hat{x}_t^{(i),+} = \begin{bmatrix} \hat{x}_{1,t}^{(i)} \\ \hat{x}_{2,t}^{(i)} \end{bmatrix}, \quad P_t^{(i),+} = \begin{bmatrix} \sigma_{\hat{x}_{1,t}}^{(i)} & 0 \\ 0 & \sigma_{\hat{x}_{2,t}}^{(i)} \end{bmatrix},$$

and

$$\hat{x}_t^{(j),+} = \begin{bmatrix} \hat{x}_{1,t}^{(j)} \\ \hat{x}_{2,t}^{(j)} \\ \hat{v}_{x_1,t}^{(j)} \\ \hat{v}_{x_2,t}^{(j)} \end{bmatrix}, \quad P_t^{(j),+} = \begin{bmatrix} \sigma_{\hat{x}_{1,t}}^{(j)} & 0 & \sigma_{\hat{x}_{1,t};\hat{v}_{x_1,t}}^{(j)} & 0 \\ 0 & \sigma_{\hat{x}_{2,t}}^{(j)} & 0 & \sigma_{\hat{x}_{2,t};\hat{v}_{x_2,t}}^{(j)} \\ \sigma_{\hat{v}_{x_1,t};\hat{x}_{1,t}}^{(j)} & 0 & \sigma_{\hat{v}_{x_1,t}}^{(j)} & 0 \\ 0 & \sigma_{\hat{v}_{x_2,t};\hat{x}_{2,t}}^{(j)} & 0 & \sigma_{\hat{v}_{x_2,t}}^{(j)} \end{bmatrix}.$$

Then by applying Lemma 2, the marginal distribution

$$\pi_{[i]}^{(j)}(x_t^{(j)}|\tilde{Y}_t^{(j)}, \tilde{U}_t^{(j)}) = \int\int \pi^{(j)}(x_t^{(j)}|\tilde{Y}_t^{(j)}, \tilde{U}_t^{(j)}) d\hat{v}_{x_1,t}^{(j)} d\hat{v}_{x_2,t}^{(j)}$$

is also normal, with mean

$$\hat{x}_{t,[i]}^{(j),+} = \begin{bmatrix} \hat{x}_{1,t}^{(j)} \\ \hat{x}_{2,t}^{(j)} \end{bmatrix}$$

and covariance matrix

$$P_{t,[i]}^{(j),+} = \begin{bmatrix} \sigma_{\hat{x}_{1,t}}^{(j)} & 0 \\ 0 & \sigma_{\hat{x}_{2,t}}^{(j)} \end{bmatrix}.$$

Lemma 2 (Marginal distributions of a multivariate normal distribution [3, 25, 26])**.** *Suppose the following mutlivariate normal random variable* $\boldsymbol{X}$

$$\boldsymbol{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \vdots \\ \mu_n \end{bmatrix}, \begin{bmatrix} \sigma_{11} & \dots & \sigma_{1n} \\ \vdots & \ddots & \vdots \\ \sigma_{n1} & \dots & \sigma_{nn} \end{bmatrix} \right)$$

with joint pdf $p_{\boldsymbol{X}}(x_1, \dots, x_n)$*. The marginal distribution of* $\boldsymbol{X}' = \left[X_{i_1}, \dots, X_{i_k}\right]^\intercal$*, where* $k \leq n$ *and* $1 \leq i_1 < \dots < i_k \leq n$*,*

$$p_{\boldsymbol{X}'}(x_{i_1}, \dots, x_{i_k}) = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} p_{\boldsymbol{X}}(x_1, \dots, x_n) dx_{j_1} \dots dx_{j_{n-k}},$$

where $1 \leq j_1 < \ldots < j_{n-k} \leq n$ *and are different from* $i_1, \ldots, i_k$*, is a multivariate normal distribution*

$$\boldsymbol{X}' = \begin{bmatrix} X_{i_1} \\ \vdots \\ X_{i_k} \end{bmatrix} \sim \mathcal{N}\Bigg(\begin{bmatrix} \mu_{i_1} \\ \vdots \\ \mu_{i_k} \end{bmatrix}, \begin{bmatrix} \sigma_{i_1 i_1} & \ldots & \sigma_{i_1 i_k} \\ \vdots & \ddots & \vdots \\ \sigma_{i_k i_1} & \ldots & \sigma_{i_k i_k} \end{bmatrix} \Bigg).$$

Putting it all together, the combination phase then proceeds as follows

$$\tilde{\pi}^{(i)}(x_t^{(i)} | \tilde{Y}_t^{(i)}, \tilde{U}_t^{(i)}) \propto \prod_{j \in \mathcal{I}_S^{(i)}} \int \underbrace{\pi^{(j)}(x_t^{(j)} | \tilde{Y}_t^{(j)}, \tilde{U}_t^{(j)})}_{\mathcal{N}(\hat{x}_t^{(j),+}, P_t^{(j),+})}{}^{a_{ij,t}} dx_{t,[j-i]}^{(j)} \tag{4.13}$$

$$= \prod_{j \in \mathcal{I}_S^{(i)}} \underbrace{\pi_{[i]}^{(j)}(x_t^{(j)} | \tilde{Y}_t^{(j)}, \tilde{U}_t^{(j)})}_{\mathcal{N}(\hat{x}_{t,[i]}^{(j),+}, P_{t,[i]}^{(j),+})}{}^{a_{ij,t}}, \tag{4.14}$$

giving us a normal distribution pdf with the following covariance matrix and mean

$$\begin{aligned} \tilde{P}_t^{(i),+} &= \Big[\sum_{j \in \mathcal{I}_S^{(i)}} a_{ij,t} \Big(P_{t,[i]}^{(j),+} \Big)^{-1} \Big]^{-1}, \\ \tilde{x}_t^{(i),+} &= \tilde{P}_t^{(i),+} \Big(\sum_{j \in \mathcal{I}_S^{(i)}} a_{ij,t} \Big(P_{t,[i]}^{(j),+} \Big)^{-1} \hat{x}_{t,[i]}^{(j),+} \Big). \end{aligned} \tag{4.15}$$

4.2.4 Filter reset

Another way nodes can exploit the information contained within their neighborhood to further improve their estimation performance is the detection of possible node failures. Node i could detect its failure by finding inconsistencies between its own estimate and the estimates of its neighbors, and whenever a failure is detected, the filter resets itself (and possibly reinitializes itself with a new initial prior estimate).

In this thesis, the following node failure detection and reset method is suggested. The node i acquires the estimates $\pi_{[i]}^{(j)}(\cdot) \equiv \mathcal{N}(\hat{x}_{t,[i]}^{(j)}, P_{t,[i]}^{(j)})$ from its neighbors $j \in \mathcal{I}_S^{(i)} \setminus \{i\}$. It then calculates the centroid $\bar{x}_t^{(i)}$ either by:

1. the arithmetic mean of $\pi_{[i]}^{(j)}(\cdot)$'s point estimates $\hat{x}_{t,[i]}^{(j)}$

$$\bar{x}_t^{(i)} = \frac{1}{|\mathcal{I}_S^{(i)}| - 1} \sum_{j \in \mathcal{I}_S^{(i)} \setminus \{i\}} \hat{x}_{t,[i]}^{(j)}, \tag{4.16}$$

2. the covariance intersection of $\pi_{[i]}^{(j)}(\cdot)$

$$\begin{aligned}\bar{P}_t^{(i)} &= \Big[\sum_{j \in \mathcal{I}_S^{(i)} \setminus \{i\}} a_{ij,t} \Big(P_{t,[i]}^{(j)} \Big)^{-1} \Big]^{-1}, \\ \bar{x}_t^{(i)} &= \bar{P}_t^{(i)} \Big(\sum_{j \in \mathcal{I}_S^{(i)} \setminus \{i\}} a_{ij,t} \Big(P_{t,[i]}^{(j)} \Big)^{-1} \hat{x}_{t,[i]}^{(j)} \Big).\end{aligned} \tag{4.17}$$

Node failure is then determined by the satisfaction of the condition

$$\|\hat{x}_t^{(i)} - \bar{x}_t^{(i)}\|_2 \geq \tau_r, \tag{4.18}$$

where τ_r is an arbitrary user-chosen threshold and $\|\cdot\|_2$ is the Euclidean norm. Provided that the above condition is satisfied, the filter is reset and reinitialized as follows:

$$\begin{aligned}\hat{x}_t^{(i)} &\leftarrow \bar{x}_t^{(i)}, \\ P_t^{(i)} &\leftarrow \frac{1}{\lambda_r^{nr}} P_0^{(i)},\end{aligned} \tag{4.19}$$

where $\lambda_r \in [0, 1]$ is similar to the forgetting factor λ in Section 4.1, nr is the number of prior resets and $P_0^{(i)}$ is the initial state estimate covariance matrix set at $t = 0$. In addition, an optional step can be used if the observation matrix $H_t^{(i)}$ is invertible and the condition (4.18) was met. The point estimate $\hat{x}_t^{(i)}$ is checked against the latest observation $y_t^{(i)}$

$$\|H_t^{(i)} \hat{x}_t^{(i)} - y_t^{(i)}\|_2 \geq \tau_r. \tag{4.20}$$

If the condition is satisfied, the filter reinitialization changes to

$$\begin{aligned}\hat{x}_t^{(i)} &\leftarrow \Big(H_t^{(i)} \Big)^{-1} y_t^{(i)}, \\ P_t^{(i)} &\leftarrow \frac{1}{\lambda_r^{nr}} \hat{P}_0^{(i)}.\end{aligned} \tag{4.21}$$

The whole procedure is summarized in Algorithm 1.

Algorithm 1: Diffusion node failure detection under partially heterogeneous models

Set the reset threshold τ_r and flattening factor λ_r. For a fixed node $i \in \mathcal{I}$ and time instant $t = 1, 2, ...$, the node i performs the following:

1. Gather estimates $\pi_{[i]}^{(j)}(\cdot)$ from the neighbors $j \in \mathcal{I}_S^{(i)} \setminus \{i\}$.
2. Compute the centroid $\bar{x}_t^{(i)}$ using either Eq. (4.16) or Eq. (4.17).
3. Evaluate the condition (4.18).
4. If satisfied, reinitialize the filter according to Eq. (4.19).
5. (Optional) If condition (4.18) was satisfied and the matrix $H_t^{(i)}$ is invertible, evaluate the condition (4.20).
6. (Optional) If satisfied, reinitialize the filter according to Eq. (4.21).

Algorithm 2: ATC diffusion Kalman filtering under partially heterogeneous models

Initialize the agents $i = 1, 2, ..., I$. Set their prior pdfs $\mathcal{N}(\hat{x}_0^{(i)}, P_0^{(i)})$ and their λ forgetting factor. Set the adaptation weights c_{ij} and combination weights a_{ij}. For $t = 1, 2, ...$ and each agent $i \in \mathcal{I}$ do:
Local prediction:

1. Flatten the prior distributions, Eq. (4.6).
2. Perform prediction of $\hat{x}_t^{(i),-}$ and $P_t^{(i),-}$, Eq. (4.8).

Failure detection:

- Reset the filter in case of node failure, Alg. 1.

Adaptation:

1. Get observations $y_t^{(j)}$ from neighbors $j \in \mathcal{I}^{(i)}$.
2. Update the conjugate prior's hyper-parameter $\xi_{t-1}^{(i)}$ by the sufficient statistics $T(y_t^{(j)})$, Eq. (3.17), or in terms of $\hat{x}_t^{(i),+}$ and $P_t^{(i),+}$, Eqs. (4.11) and (4.10).

Combination:

1. Acquire posterior pdfs from neighbors $j \in \mathcal{I}_S^{(i)}$.
2. Combine the acquired posterior pdfs according to (3.21), or in terms of $\tilde{x}_t^{(i),+}$ and $\tilde{P}_t^{(i),+}$, Eq. (4.15).

Chapter 5

Simulation experiments

The performance of the proposed diffusion filter is evaluated on the following simulation examples. The nodes of the diffusion network observe a noisy realization of a 2D trajectory. The data is simulated from the initial state $x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$ using the CAM, as described in Section 1.3.3, with $\hat{q}_{CAM} = 5 \cdot 10^{-4}$, $\Delta_t = 1$, and the time-invariant observation noise matrix $R = 25^2 \cdot I_{2\times 2}$, while R is the same for all nodes, each node receives a different set of measurements. The nodes employ one of the following state-space models in their estimation:

1. constant acceleration model, Sec. 1.3.3, with the parameters:

$$\begin{aligned} \hat{x}_0 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}, \\ P_0 &= 1000 \cdot I_{6\times 6}, \\ \hat{q}_{CAM} &= 5 \cdot 10^{-4}, \\ \lambda_{CAM} &= 1.00, \end{aligned}$$

2. constant velocity model, Sec. 1.3.2, with the parameters:

$$\begin{aligned} \hat{x}_0 &= \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}, \\ P_0 &= 1000 \cdot I_{4\times 4}, \\ \hat{q}_{CVM} &= 10 \cdot \hat{q}_{CAM}, \\ \lambda_{CVM} &= 0.95, \end{aligned}$$

3. random-walk model, Sec. 1.3.1, with the parameters:

$$\begin{aligned} \hat{x}_0 &= \begin{bmatrix} 0 & 0 \end{bmatrix}^\mathsf{T}, \\ P_0 &= 1000 \cdot I_{2\times 2}, \\ \hat{q}_{RWM} &= 20 \cdot \hat{q}_{CAM}, \\ \lambda_{RWM} &= 0.90, \end{aligned}$$

where $I_{d\times d}$ is a $d \times d$ identity matrix. The threshold for reset was set at $\tau_r = 10.0$. For the node $i \in \mathcal{I}$, the adaptation weights are

$$c_{ij} = \begin{cases} 1 & j \in \mathcal{I}^{(i)}, \\ 0 & j \notin \mathcal{I}^{(i)}, \end{cases}$$

and the combination weights are

$$a_{ij} = \begin{cases} \frac{1}{|\mathcal{I}_S^{(i)}|} & j \in \mathcal{I}_S^{(i)}, \\ 0 & j \notin \mathcal{I}_S^{(i)}. \end{cases}$$

The results are averaged over 100 independently simulated trajectories for $t = 1, 2, ..., 1000$. Figure 5.1 shows one such simulated trajectory.

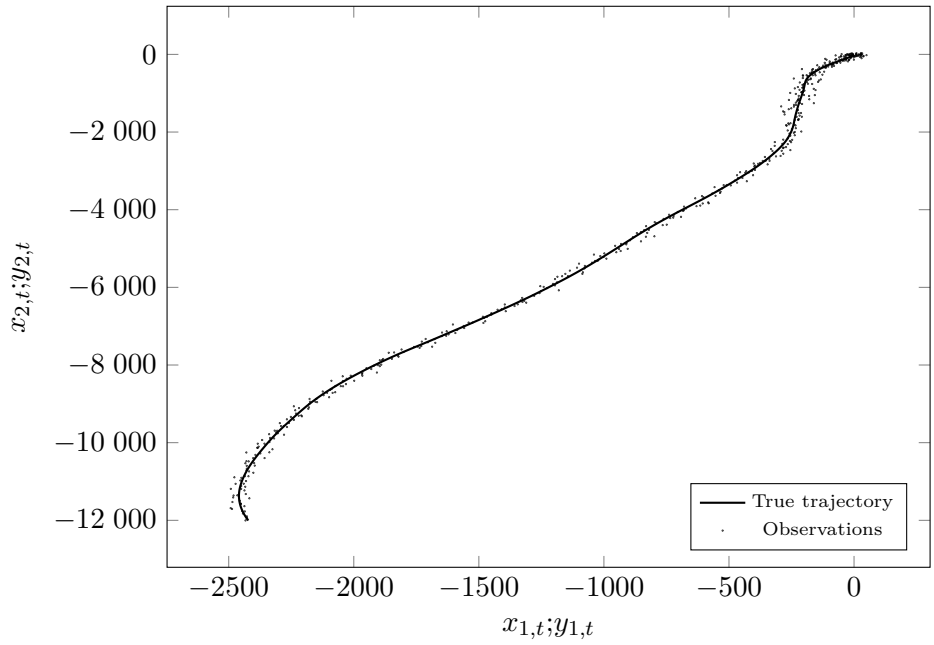

Figure 5.1: An example of one simulated trajectory, only the first 500 steps are depicted.

5.1 Example 1: Fixed initialization of nodes

The estimation is run on a network with $|\mathcal{I}| = 15$ nodes, with node degrees ranging from 4–8 (the average is 6.14). The network consists of 2 RWM nodes, 4 CVM nodes, and 9 CAM nodes. In this example, the network structure stays fixed between consecutive simulation runs. The network is pictured in Figure 5.2. The following four strategies were tested: (i) no cooperation,

(ii) adapt-then-combine (ATC) without reset, (iii) ATC with reset, centroid $\bar{x}$ calculated as the arithmetic mean, Eq. (4.16), and (iv) ATC with reset, centroid $\bar{x}$ calculated using covariance intersection, Eq. (4.17), the scenarios implementing the reset strategy are summarized in Algorithm 2.

Figures 5.3, 5.4, 5.5 and 5.6 show the evolution of RMSE of nodes utilizing the same type of model for the four tested strategies. In the no cooperation and, to a lesser extent, ATC-only scenario, the RWM nodes cannot keep up with the changes in position, caused by the growing velocity and acceleration, between successive time instants, hence, causing an early filter divergence. The resetting strategies help the RWM nodes achieve an estimation performance similar to that of the better models, though, at the cost of a higher communication burden. The performance difference of the two resetting strategies seems insignificant, at least with the current simulation configuration. Figures 5.7, 5.8 and 5.9 show that the resetting strategies seem to only significantly help the RWM nodes, the performance of CVM and CAM nodes seems largely unaffected. State and state estimate evolution of a few selected cases are shown in Figures 5.10, 5.11, 5.12 and 5.13. Figure 5.12 is of particular interest, as it reveals that after a certain point, even the RWM filter with reset just cannot keep up with the growing changes in position between successive measurements caused by velocity and acceleration, and keeps resetting every iteration.

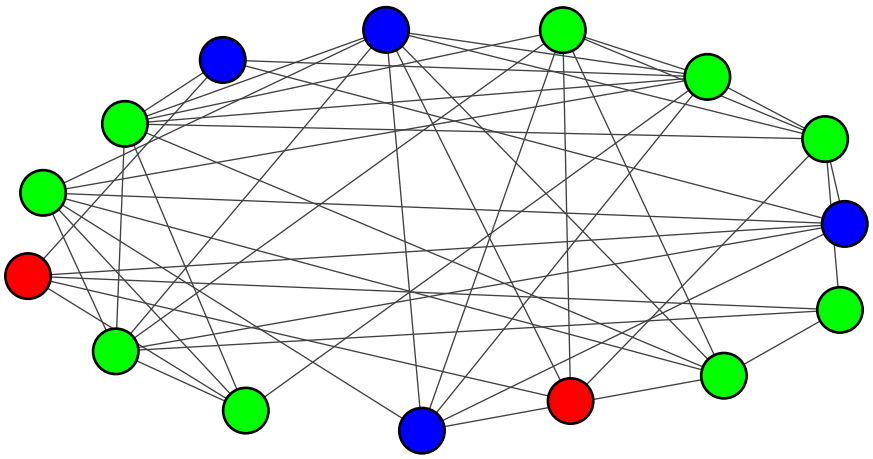

Figure 5.2: The network topology used in the simulation - 2 RWM nodes (red), 4 CVM nodes (blue) and 9 CAM nodes.

5.2 Example 2: Random initialization of nodes

The second example utilizes the same network topology seen in Figure 5.2. The number of RWM, CVM and CAM agents stays the same. However,

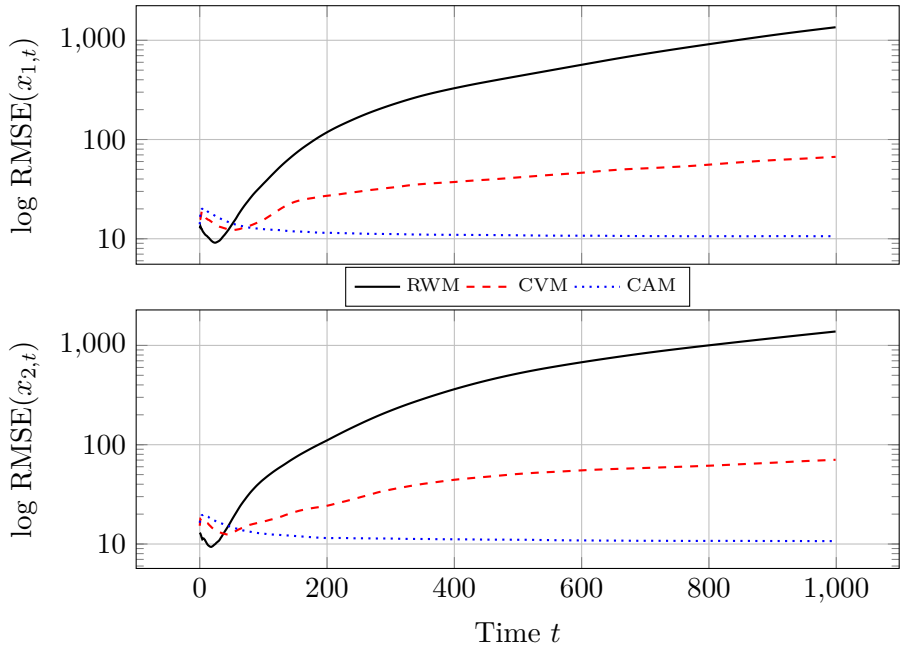

Figure 5.3: NOCOOP: Fixed initialization. Average RMSE of nodes employing the same model, averaged over 100 simulations.

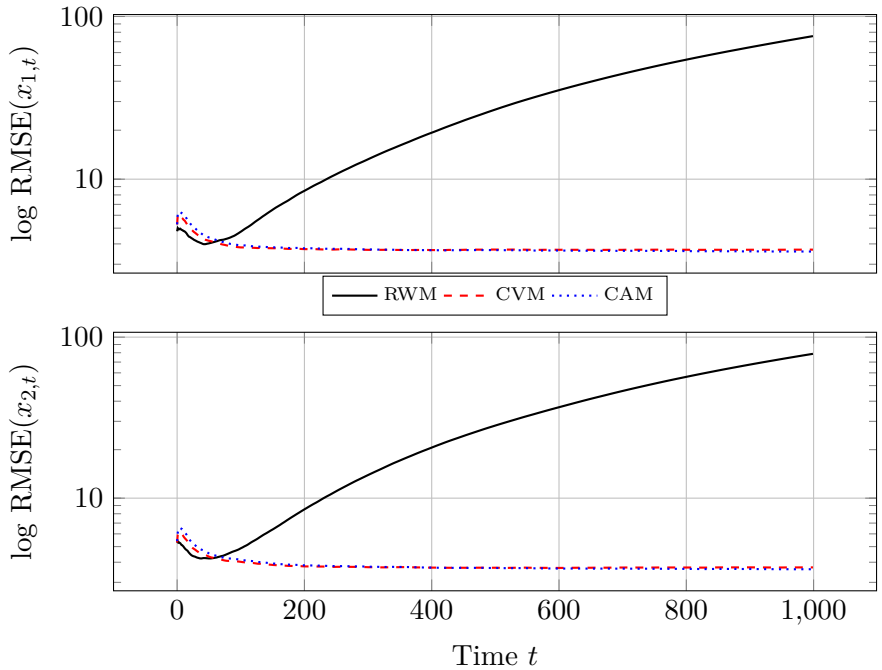

Figure 5.4: ATC(Noreset): Fixed initialization. Average RMSE of nodes employing the same model, averaged over 100 simulations.

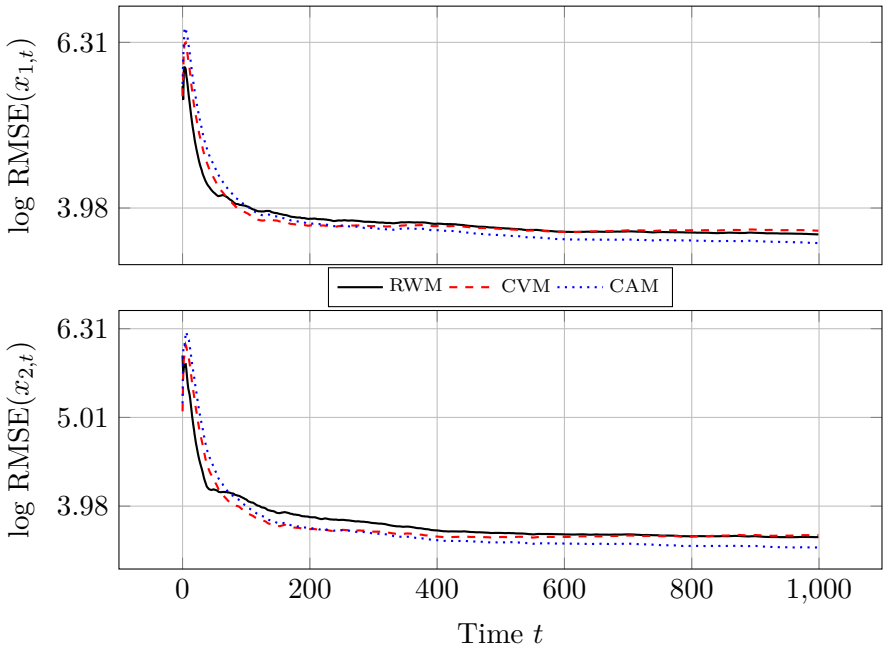

Figure 5.5: ATC+Reset(Amean): Fixed initialization. Average RMSE of nodes employing the same model, averaged over 100 simulations.

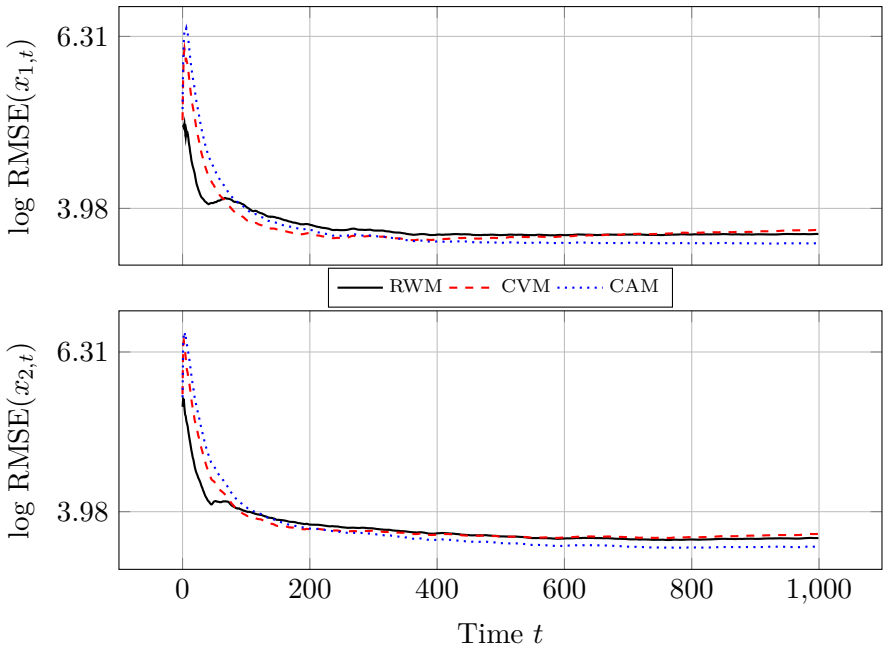

Figure 5.6: ATC+Reset(CI): Fixed initialization. Average RMSE of nodes employing the same model, averaged over 100 simulations.

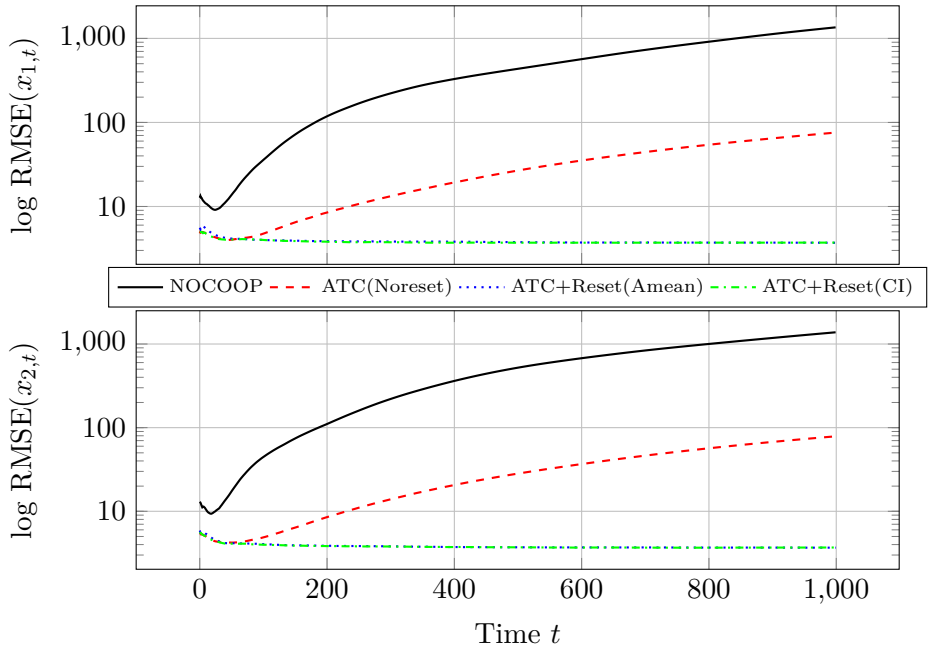

Figure 5.7: Fixed initialization. Average RMSE of RWM nodes, comparison of strategies.

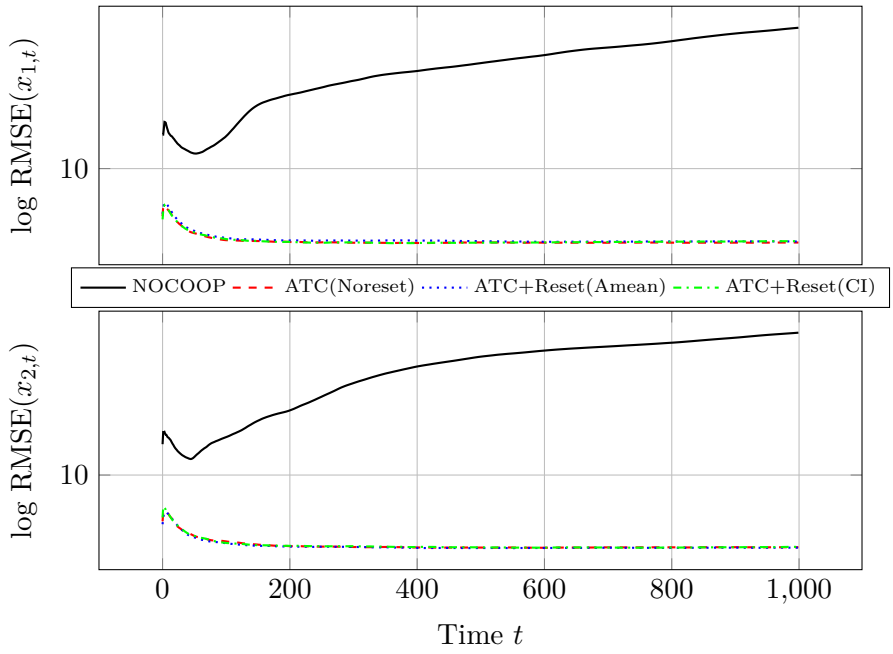

Figure 5.8: Fixed initialization. Average RMSE of CVM nodes, comparison of strategies.

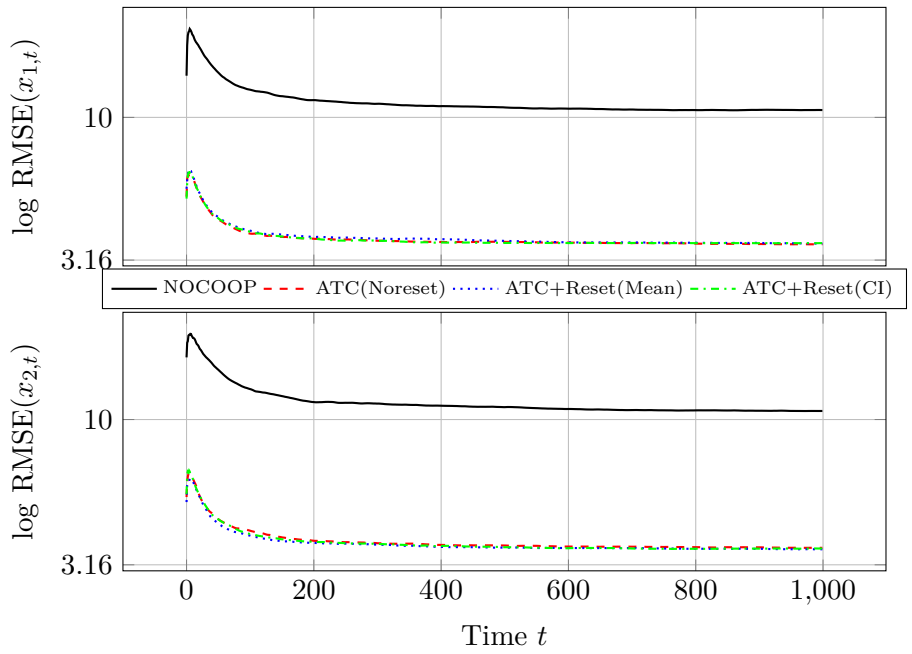

Figure 5.9: Fixed initialization. Average RMSE of CAM nodes, comparison of strategies.

unlike the first example, where the employed state-space models stay fixed across simulation runs, here the models are randomly assigned to the nodes in each simulation run.

We can directly compare Figures 5.4 vs. 5.15, Figures 5.5 vs. 5.16, and 5.6 vs. 5.17 to see the performance difference between fixed versus random initialization of nodes. In the first case, comparison of ATC-only scenarios, it seems like the structure of the network in the first example causes divergence of the RWM filters at a faster rate (note the values on the y-axis). Interestingly, in the reset strategies averaged over 100 random initializations, the RWM nodes seem to sometimes slightly outperform the more complex models. What might be happening is that in some network configurations, the underspecified models actually lower the efficiency of diffusion of information over the network. This happens in the combination phase due to the one-way communication between nodes of different levels of complexity. The lower complexity nodes do not share this information further to their other neighbors if they are of higher complexity, creating a sort of communication "black hole", an extreme example can be seen in Fig. 5.14. Figure 5.18 shows the state and state estimate evolution for a selected RWM node, unlike Fig. 5.12, the "constant resetting at every time step" seems to be delayed, but it still happens at around $t = 200$.

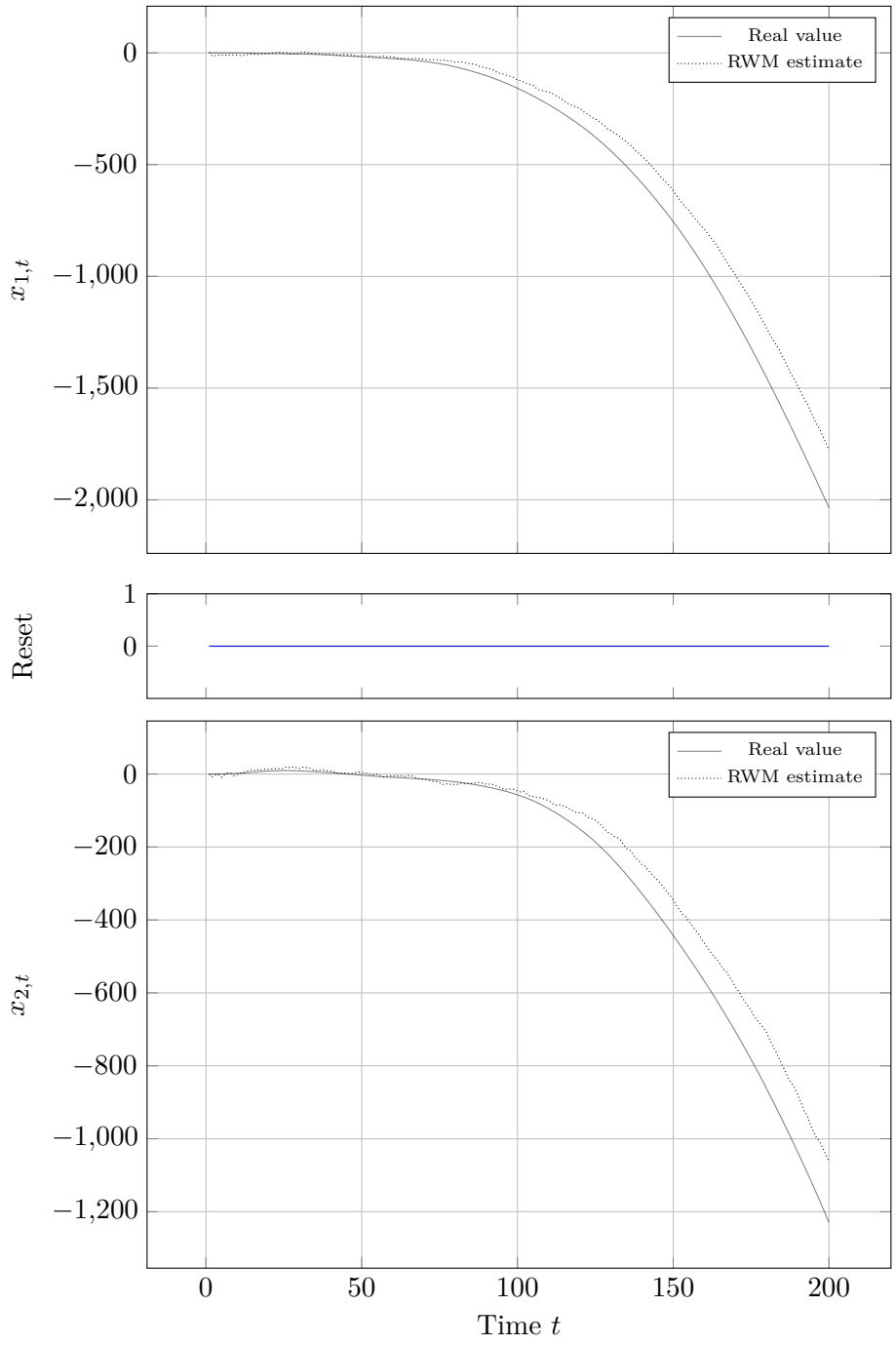

Figure 5.10: Evolution of states and state estimates of an isolated RWM node. Only the first 200 steps are shown to emphasize the differences. (Example 1)

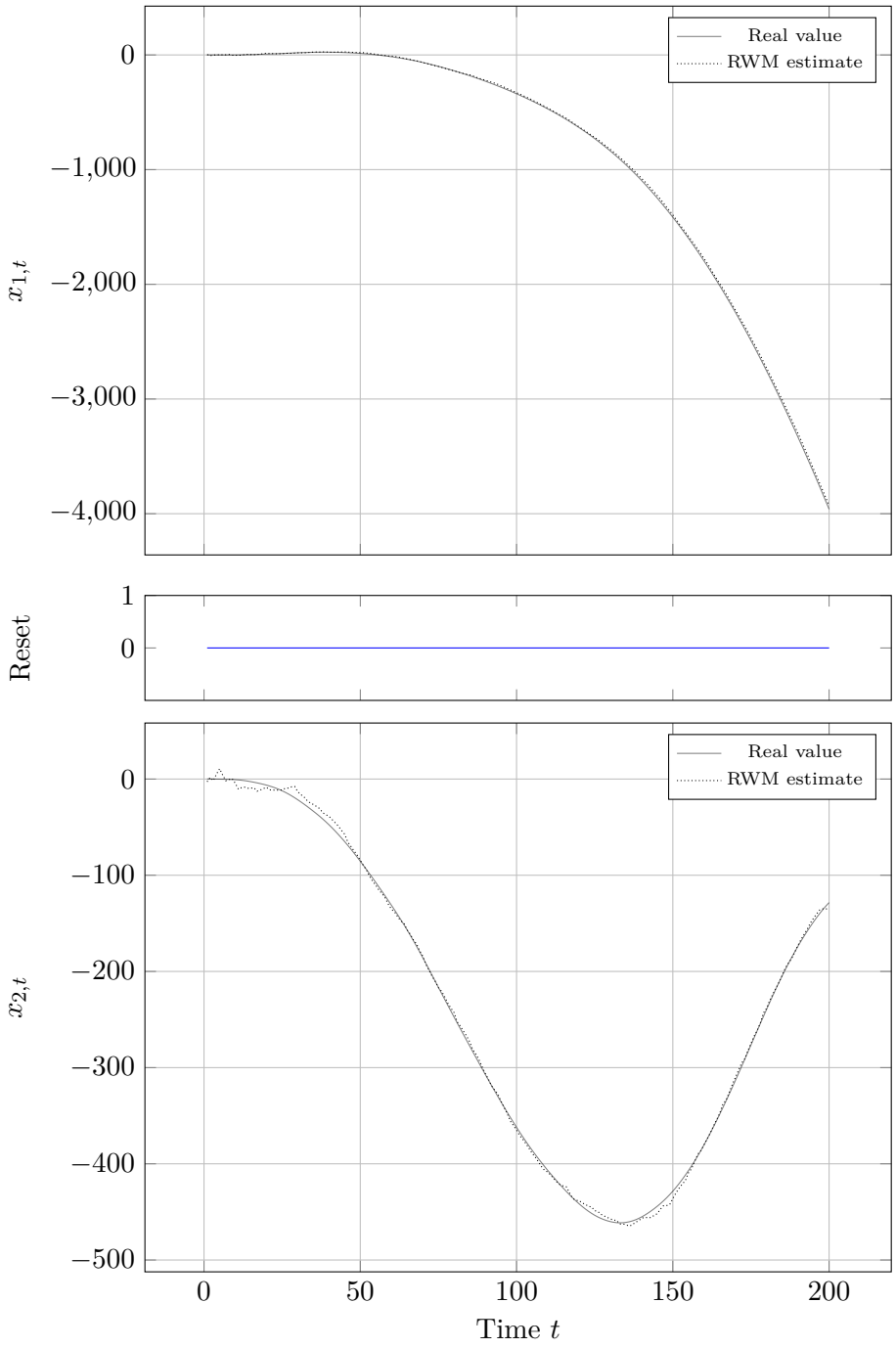

Figure 5.11: Evolution of states and state estimates of a selected RWM node using the ATC strategy without reset. Only the first 200 steps are shown to emphasize the differences. (Example 1)

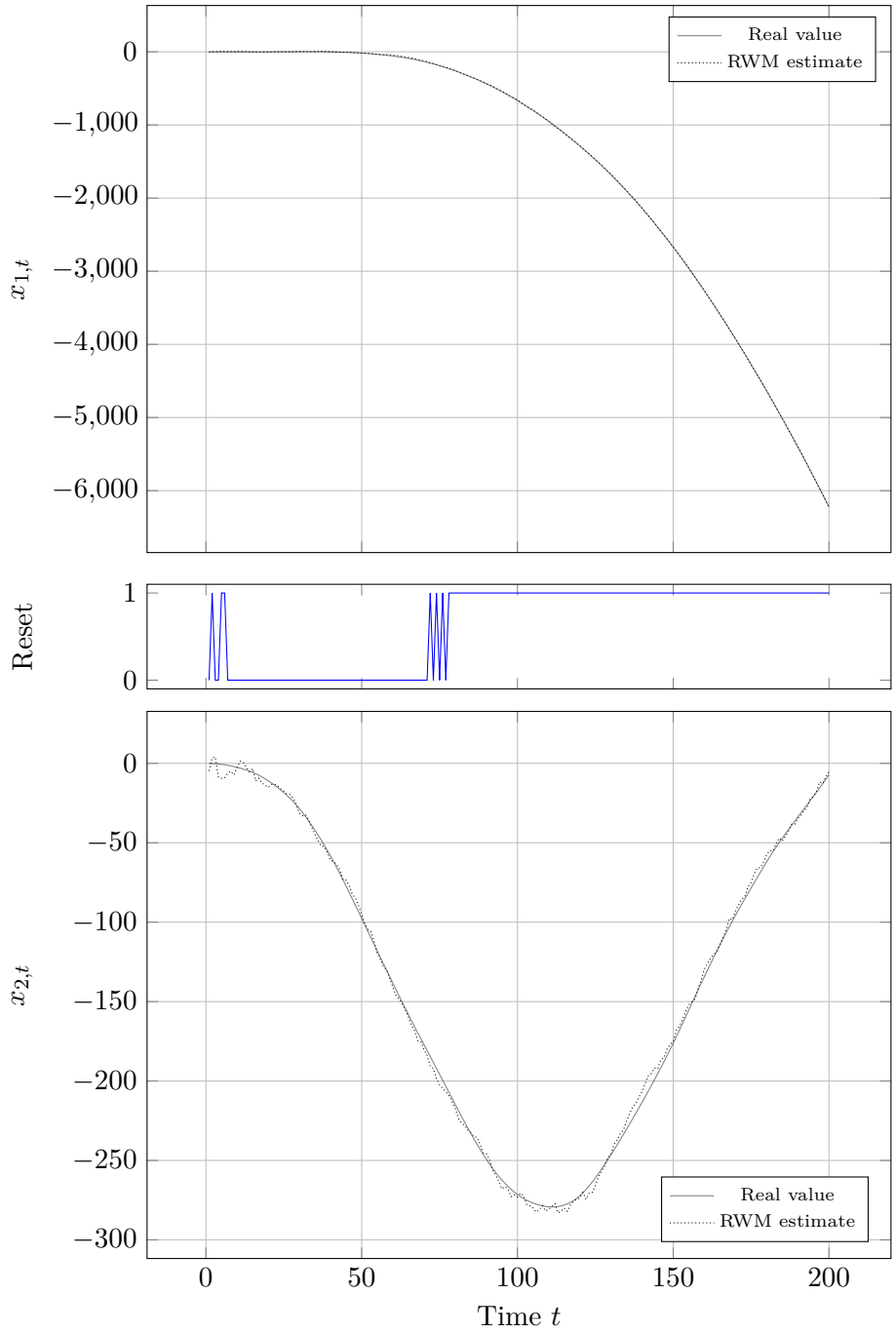

Figure 5.12: Evolution of states and state estimates of a selected RWM node using the ATC strategy with reset based on distance from the arithmetic mean centroid, Eq. (4.16). Only the first 200 steps are shown to emphasize the differences. (Example 1)

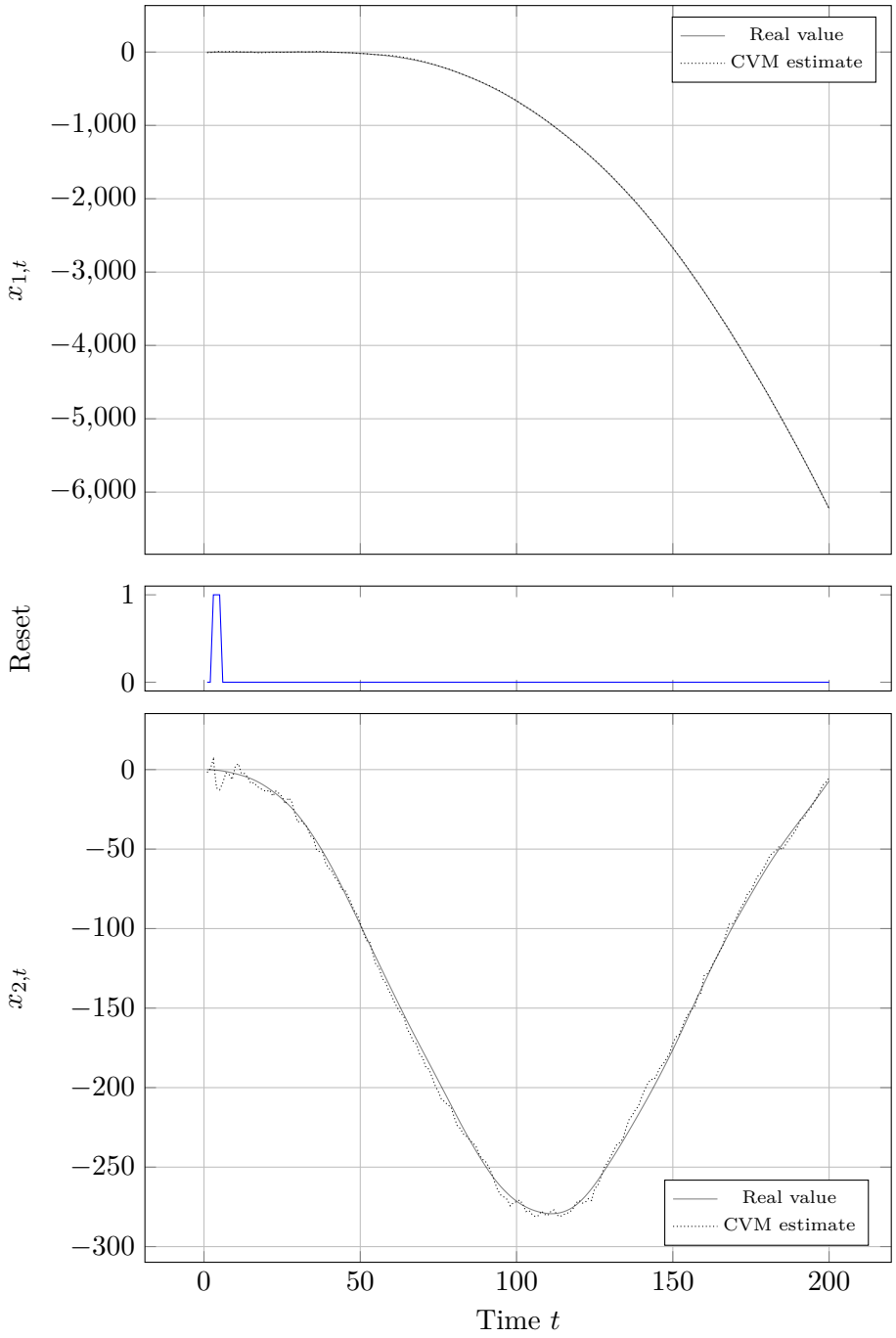

Figure 5.13: Evolution of states and state estimates of a selected CVM node using the ATC strategy with reset based on distance from the arithmetic mean centroid, Eq. (4.16). Only the first 200 steps are shown to emphasize the differences. (Example 1)

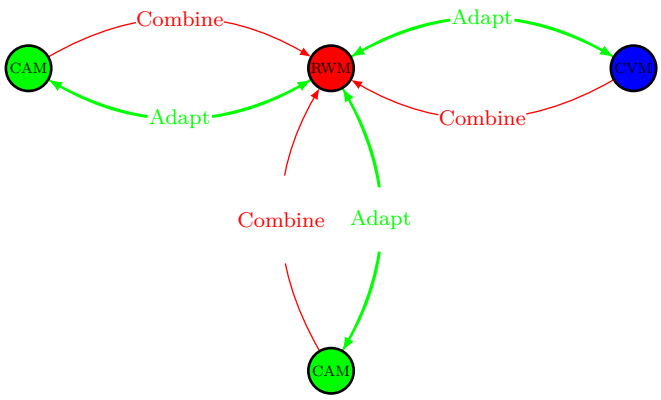

Figure 5.14: RWM node causing a communication bottleneck for combination/reset

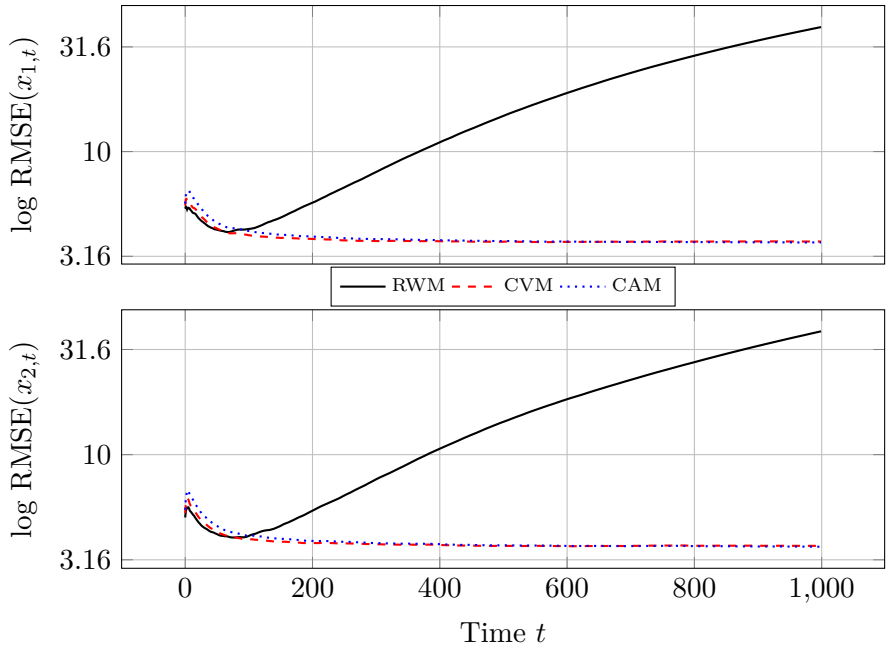

Figure 5.15: ATC(Noreset): Random initialization. Average RMSE of nodes employing the same model, averaged over 100 simulations.

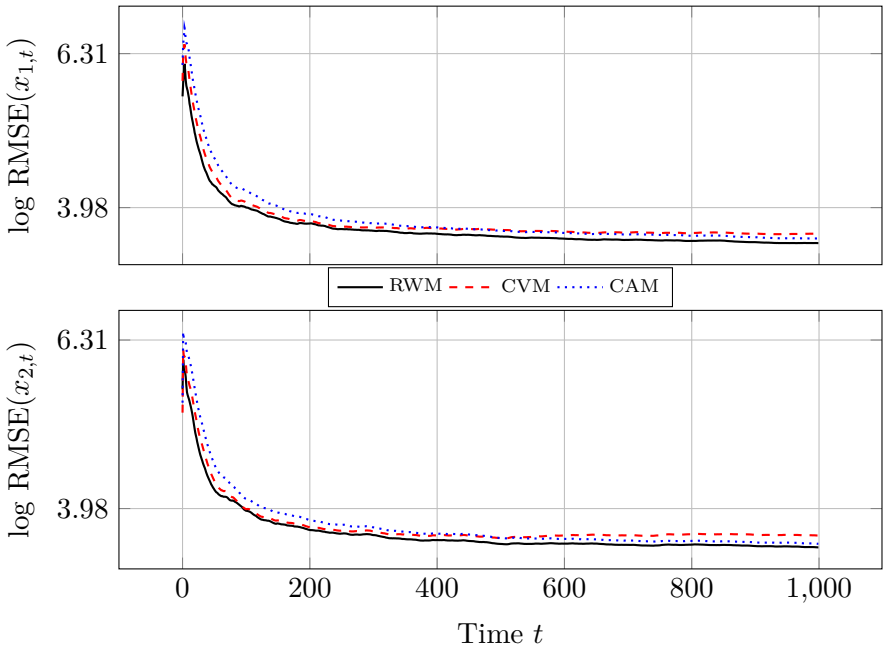

Figure 5.16: ATC+Reset(Amean): Random initialization. Average RMSE of nodes employing the same model, averaged over 100 simulations.

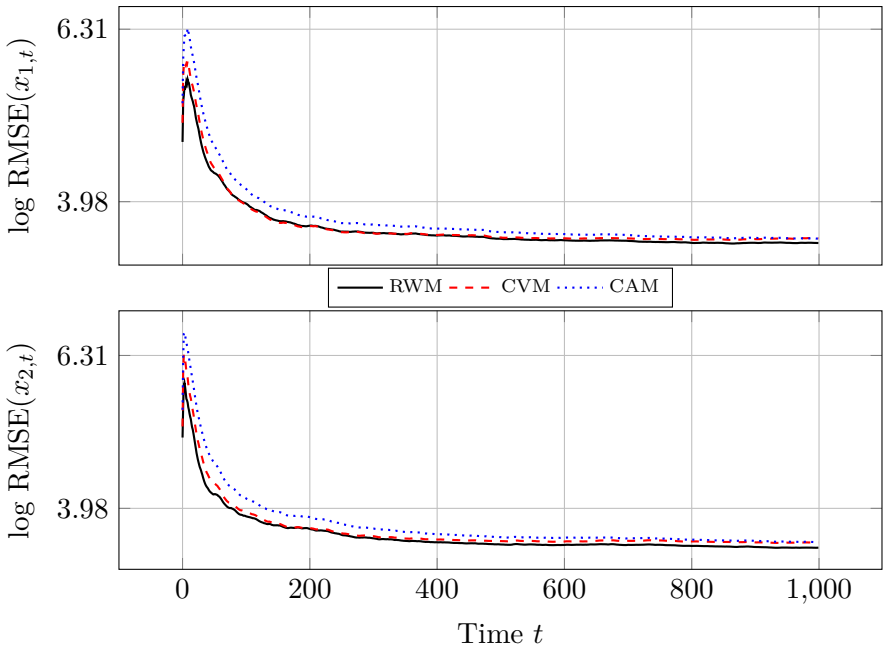

Figure 5.17: ATC+Reset(CI): Random initialization. Average RMSE of nodes employing the same model, averaged over 100 simulations.

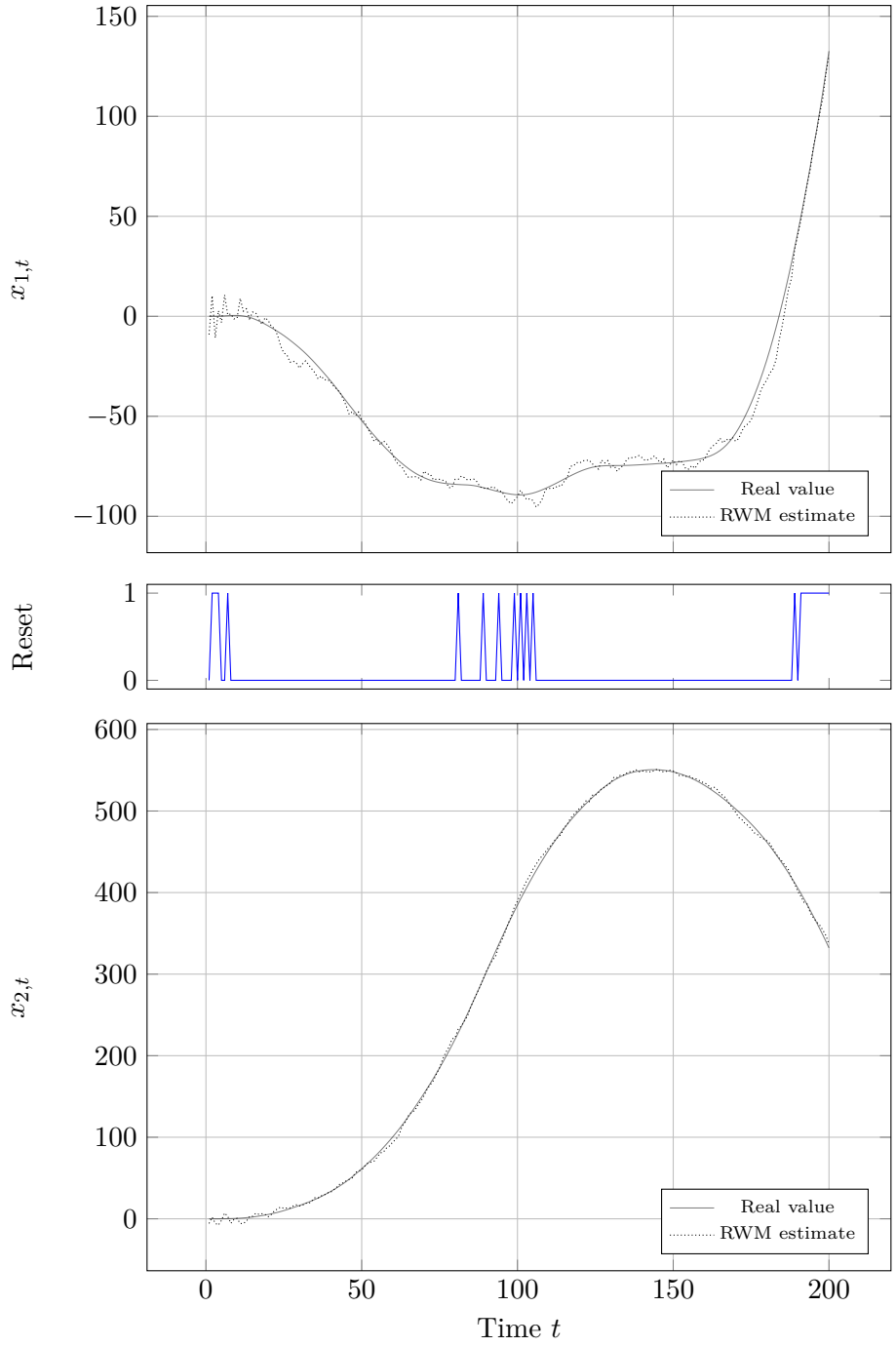

Figure 5.18: Evolution of states and state estimates of a selected RWM node using the ATC strategy with reset based on distance from the arithmetic mean centroid, Eq. (4.16). Only the first 200 steps are shown to emphasize the differences. (Example 2)

Conclusion

The objective of this thesis was to examine the problem of distributed Kalman filtering under partially heterogeneous models and propose a collaboration method, which would improve the estimation performance and stability of the underspecified models. Another goal was to test the performance of the suggested method on a concrete example: tracking a moving target in a 2D space using a network of random-walk model, constant velocity model, and constant acceleration model nodes.

Chapter 1 introduces the linear state-space model and the Bayesian approach to its estimation. In Chapter 2 a brief overview of the available distributed estimation strategies is given, but later focuses mostly on the adapt-then-combine (ATC) diffusion strategy. Chapter 3 then explores the renowned Kalman filter and its distributed diffusion counterpart, deriving both of them using theory from the previous chapters. In Chapter 4 the problem of filtration under partially heterogeneous models is examined, and the issues of the underspecified models are discussed. A modification to the combination phase of the diffusion Kalman filter is proposed, providing the underspecified models a way to extract and incorporate compatible information from the more complex models. Lastly, a simple node failure detection method is proposed to further improve the performance of the underspecified models. Finally, Chapter 5 assesses the performance of the proposed methods on a couple of simulation examples. Overall, the results suggest an improved performance of the underspecified nodes compared to the no cooperation scenario. The failure detection and reset mechanism further improves performance of the underspecified models, although it does add an additional communication overhead. A possible downside of the proposed methods is also mentioned — the inclusion of underspecified models in the network may act as a communication bottleneck, slowing down the diffusion of information through the network.

Improvements and future work

In Chapter 5, the effects of, e. g., network size, node degree, and number of RWM/CVM/CAM nodes could have been explored. Besides that, instead of only uniform combination weights a_{ij}, a more elaborate strategy, such as the one described in [8, Chap. V.] could have been implemented.

Future work could involve filtering under unknown heterogeneous noise covariance matrices [4], implementation of an adaptive filter to deal with inaccurate or unknown covariance matrices [27], or utilization of more sophisticated combination strategies [28, 29]. Another interesting question to explore — is there a way for the agents to improve their performance by incorporating information from their neighbors employing less complex models? Which information to incorporate and how?

Bibliography

[1] F. S. Cattivelli and A. H. Sayed, "Diffusion Strategies for Distributed Kalman Filtering and Smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.

[2] J. Hu, L. Xie, and C. Zhang, "Diffusion Kalman Filtering Based on Covariance Intersection," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 891–902, Feb. 2012.

[3] K. Dedecius, "Diffusion estimation of state-space models: Bayesian formulation," in *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2014, pp. 1–6, iSSN: 2378-928X.

[4] K. Dedecius and O. Tichý, "Collaborative Sequential State Estimation Under Unknown Heterogeneous Noise Covariance Matrices," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5365–5378, 2020.

[5] K. Bar-Shalom and X.-R. L. Li, *Estimation with applications to tracking navigation*. John Wiley & Sons, Jun. 2001.

[6] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.

[7] G. Kitagawa and W. Gersch, *Smoothness priors analysis of time series*. New York, NY: Springer, 1996, vol. 116.

[8] K. Dedecius and P. M. Djurić, "Sequential estimation and diffusion of information over networks: A bayesian approach with exponential family of distributions," *IEEE Transactions on Signal Processing*, vol. 65, pp. 1795–1809, 2017.

[9] H. Raiffa and R. Schlaifer, *Applied statistical decision theory.* Boston: Division of Research, Graduate School of Business Adminitration, Harvard University, 1961.

[10] K. Dedecius and V. Seckarova, "Dynamic Diffusion Estimation in Exponential Family Models," *IEEE Signal Processing Letters*, vol. 20, no. 11, pp. 1114–1117, nov 2013.

[11] S. He, H.-S. Shin, S. Xu, and A. Tsourdos, "Distributed estimation over a low-cost sensor network: A Review of state-of-the-art," *Information Fusion*, vol. 54, pp. 21–43, Feb. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253518307747

[12] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications - WSNA '02*, ser. WSNA '02. New York, NY, USA: ACM Press, 2002, p. 88–97.

[13] T. T. Vu and A. R. Rahmani, "Distributed Consensus-Based Kalman Filter Estimation and Control of Formation Flying Spacecraft: Simulation and Validation," in *AIAA Guidance, Navigation, and Control Conference*, ser. AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, Jan. 2015.

[14] A. H. Sayed, "Adaptation, Learning, and Optimization over Networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014. [Online]. Available: http://infoscience.epfl.ch/record/233482

[15] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed Kalman filtering based on consensus strategies," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 622–633, 2008.

[16] R. Olfati-Saber, "Kalman-Consensus Filter : Optimality, stability, and performance," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 7036–7042.

[17] S. Kar and J. M. F. Moura, "Gossip and Distributed Kalman Filtering: Weak Consensus Under Weak Detectability," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1766–1784, Apr. 2011.

[18] K. Ma, S. Wu, Y. Wei, and W. Zhang, "Gossip-Based Distributed Tracking in Networks of Heterogeneous Agents," *IEEE Communications Letters*, vol. 21, no. 4, pp. 801–804, Apr. 2017.

[19] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory.* John Wiley & Sons, Inc., may 1994.

[20] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons, Jun. 2006.

[21] V. Peterka, "BAYESIAN APPROACH TO SYSTEM IDENTIFICATION," in *Trends and Progress in System Identification*. Elsevier, 1981, pp. 239–304.

[22] H. Masnadi-Shirazi, A. Masnadi-Shirazi, and M.-A. Dastgheib, "A Step by Step Mathematical Derivation and Tutorial on Kalman Filters," Oct. 2019. [Online]. Available: https://arxiv.org/abs/1910.03558

[23] S. J. Julier and J. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, vol. 4, Jun. 1997, pp. 2369–2373 vol.4.

[24] K. Dedecius, I. Nagy, and M. Kárný, "Parameter tracking with partial forgetting method," *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 1, pp. 1–12, aug 2011.

[25] P. Hrabák, D. Vašata, and P. Novák, "NI-VSM: Course materials," 2012–2021. [Online]. Available: https://courses.fit.cvut.cz/MI-SPI/index.html

[26] R. Johnson, *Applied multivariate statistical analysis*. Upper Saddle River, N.J: Pearson Prentice Hall, 2007.

[27] Y. Huang, Y. Zhang, Z. Wu, N. Li, and J. Chambers, "A Novel Adaptive Kalman Filter With Inaccurate Process and Measurement Noise Covariance Matrices," *IEEE Transactions on Automatic Control*, vol. 63, pp. 594–601, 2018.

[28] D. Jin, J. Chen, C. Richard, J. Chen, and A. H. Sayed, "Convex Combination of Diffusion Strategies Over Networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 714–731, 2020.

[29] ——, "Affine Combination of Diffusion Strategies Over Networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2087–2104, 2020.

Appendix A

Acronyms

ATC adapt-then-combine.

CAM constant acceleration model.

CTA combine-then-adapt.

CVM constant velocity model.

HMM hidden Markov model.

KLD Kullback-Leibler divergence.

MCMC Markov chain Monte Carlo.

pdf probability density function.

RMSE root mean square error.

RWM random-walk model.

Appendix B

Contents of enclosed CD

- `readme.txt` ..... the file with CD contents description
- `src` ..... the directory of source codes
  - `thesis` ..... the directory of LaTeX source codes of the thesis
  - `simulations` .... the directory of simulation source codes of the thesis
- `text` ..... the thesis text directory
  - `thesis.pdf` ..... the thesis text in PDF format