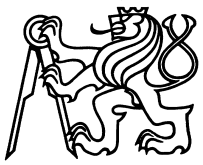
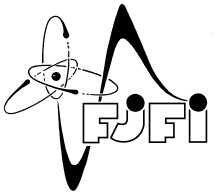


Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering



Distributed Kalman filtration under unknown spatially heterogeneous noise

Master Thesis

Author: Bc. Daniel Hnyk

Supervisor: Ing. Kamil Dedecius, Ph.D.

Academic year: 2017/2018

Acknowledgment: add ack

Declaration:

I hereby certify that this text represents my own work and that all used sources and materials are listed in the bibliography.

Prague, Saturday 21st April, 2018 Daniel Hnyk

Title: Distributed Kalman filtration under unknown spatially heterogeneous noise

Author: Bc. Daniel Hnyk

Branch of study: Mathematical Informatics

Abstract:
add abstract

Key words: Bayesian inference, Kalman filter

Contents

Introduction

A rapid development in information technology hardware over last two decades boosted affordability of cheap, yet powerful devices capable of carrying out complex computations. Variety of available sensors, which these devices can be equipped with, is overwhelming. Omnipresent smartphones or cars contains hundreds of sensors constantly collecting data and decide on life-saving or life-threatening actions. Surveillance systems, composed of hundreds or even thousands of interconnected measurement nodes keep nationwide security in place, always prepare to act in seconds based on the latest data. The so-called Internet of Things, many small computers connected to each other is only an another example of this phenomenon. At the same time, the environments where these devices operate often suffer from the imperfections of their measurements or computations. These may be caused by either misalignment of their models with a reality or by noisy random fluctuations in their operations.

For the sake of getting valuable insights from the collected data of these noisy data collection and evaluation networks, efficient processing algorithms must be developed. There is a long history of research in so called consensus algorithms as well as in the noise filtering methods. While the former try to solve fundamental problems connected to distributed computing and multi-agent systems of agreeing on something, the latter deals with the always-present noise in the received data. What is worse, many methods are sensitive to the type of the noise, making them irrelevant in some use cases.

Not being able to effectively filter the noise from the estimates makes the data less useful than it otherwise could be. The same is true the estimate redundancy is not exploited. One of notable approaches trying to solve this problem is Distributive Kalman Filtering, an extension of a hugely popular filter, heavily used since its formulation in 1960. This work is concerned with one particular instance of the Distributive Kalman Filtering, using state-of-the-art techniques and latest research in probability theory. It combines a specific probabilistic variant of the classic Kalman Filter with a recently developed so-called parameter diffusion. Both techniques leverage from the fact that can be fully expressed in Bayesian probability terms and seems to be an excellent candidate as a solution for the problem of combining information from multiple agents in noisy environment. The following pages are describing the development of such an algorithm and explain why it may be a suitable solution and under what conditions. Additionally, real implementation in Python language is used to run various simulations.

The work is divided into theoretical and practical part. The former states all the necessary theoretical background for the latter. In the first chapter, Bayesian probability is introduced as a building block for Variational Bayes inference. This is a general technique which serves as the crux for the proposed algorithm internals. The second chapter is concerned with Kalman filtering and its distributed extension. General introduction with brief applications overview is followed by mathematical formulation and discussion of its key properties. The third part is a connection of these components – Variational Bayes and Distributed Kalman filtering. The derivation of the mathematical formulation using exponential family form is carried out and in the end, a general, step-by-step algorithm is outlined. The final chapter containing the results firstly comments

on implementation details including the ecosystem. Subsequently, simulations and comparison of different variants are performed, trying to evaluate the algorithm performance, explaining an observed behavior and pinpointing its properties, features and disadvantages. Each of the chapters contain an introductory text outlining what is going to be presented and if necessary, establishes the minimal requirements.

Chapter 1

Bayesian probability

In this chapter, I summarize basics and preliminaries of Bayesian interpretation of probability and shortly compare it to frequentist approach. Reader is expected to know basic concepts from probability such as probability event, space, measure, product and sum rule, conditional and joint probability. All of these can be found in [1].

1.1 Probability interpretations

As it is a case with other branches of mathematics, probability theory has multiple interpretations with theirs own strengths and weaknesses. In this particular case, it is a subject of dispute if the term "interpretations" should be used, since there is no single formal system "probability". For example, some of the leading interpretations fail to satisfy the most common Kolmogorov's axiomatization and they still prove themselves useful[2]. Other axiomatization has been proposed for different interpretation [3] and some of the leading interpretations such as [1] do not even derive from mathematical axioms, but rather somewhat vague statements:

1. representations of plausibility are to be given by real numbers
2. plausibilities are to be in qualitative agreement with common sense
3. the plausibilities are to be "consistent", in the sense that anyone with the same information would assign the same real numbers to the plausibilities.

Interpretations don't differ only in the underlying set of axioms, but also in their practical usage or their epistemological status. Mathematicians as well as philosophers has been tackling these issues at least for over five centuries.

It's hard to fully appreciate and understand main concepts and advantages of Bayesian reasoning if one is not familiar with other interpretations, such as frequentist. Its full description can be found in [4], here we'll cover just the basics arising from the comparison.

The main idea of Bayesian interpretation is its subjective nature, where there is no such thing as *real* probability in the world, but probabilities are in actor's minds. Probability then represents agent's *state of knowledge* or *degree of belief* [5, 6, 7]. This is not a negligible formal difference – it has important consequences, such as the fact that all agents with different state of knowledge can assign different probabilities to the same event. In Bayesian framework, one explicitly convey uncertainty in his models and objects (such as random variables).

Another important aspect worth mentioning is also the fact that Bayes interpretation allows doing predictions for events which haven't occurred yet (where frequentists are unable to do anything). This property is also connected to a fact that one can do partial (*online*) updates based on new evidence.

1.2 Principles of Bayesian probability

Core idea of Bayesian interpretation is surely a famous Bayes theorem (1.1), which is easily obtained from definition of conditional probability [8].

Theorem 1. Let p be a probability measure, $\mathbf{A}$ and $\mathbf{B}$ are events, $p(\mathbf{B}) \neq 0$. Then

$$p(\mathbf{A}|\mathbf{B}) = p(\mathbf{A}) \cdot \frac{p(\mathbf{B}|\mathbf{A})}{p(\mathbf{B})} \tag{1.1}$$

where:

1. $p(\mathbf{A})$ and $p(\mathbf{B})$ are the probabilities of observing A and B independently.
2. $p(\mathbf{A}|\mathbf{B})$ is the probability of observing event A given that B is true.
3. $p(\mathbf{B}|\mathbf{A})$ is the probability of observing event B given that A is true. This factor is called **likelihood**.

To relate this simple formula to previous section, consider a scenario where one wants to use Bayesian framework for *inference.* Suppose that $\mathbf{A}$ is a hypothesis whose probability can be affected by some evidence and $\mathbf{B}$ is an evidence for this hypothesis. $p(\mathbf{A})$ in formula (1.1) can then be seen as a **prior probability** – a probability one would assign to $\mathbf{A}$ is true *before* observing the evidence. Factor $p(\mathbf{B}|\mathbf{A})$, the *likelihood*, represents how likely we expect to observe $\mathbf{B}$ *given* that $\mathbf{A}$ is true. $p(\mathbf{B})$ is same for all possible hypothesis and serves as a normalizing factor (that is the reason why is often left out in many computations). And finally, $p(\mathbf{A}|\mathbf{B})$, which is the final **posterior probability** which is the factor one usually care about – the probability of $\mathbf{A}$ after observing $\mathbf{B}$.

A factor $\frac{p(\mathbf{B}|\mathbf{A})}{p(\mathbf{B})}$ has also quite useful interpretation and that is the overall impact of evidence $\mathbf{B}$ on the probability of $\mathbf{A}$.

1.3 Statistical model

Chosen interpretation has a nontrivial influence on what such concept as *statistical model* is and how *inference* is done. I will follow traditional approach taken in [9, 10] and extend it of some necessary concepts from [11].

A statistical model is a pair of (S, M), where S is a set of possible observations (*sample space*) and M is a set of probability distributions on S. Distribution in M are expected to be approximately close to the "true" distribution which generates the data[1]. One of the main distinctions which can be made to separate families of statistical models is how distributions in M are described:

1. **Parametric models**: in this case, the set M is parameterized by some parameter θ and can have values from *finite* parameter space$\Theta \subseteq \mathbb{R}^d$. In other words: $M = \{P_\theta | \theta \in \Theta\}$.
2. **Nonparametric models**: this family is an unfortunate misnomer – nonparametric models also have parameters, but they differs from parametric models in a way that the parameter space is Θ is *infinite* – it is not fixed and can grow with the amount of data.

[1] There is a saying: "All models are wrong, but some of them are useful"

In Bayesian setting, parameters of the model have some prior distributions assigned. These distributions capture prior knowledge one posses before the application. For instance, this can be based on previous research or reliable enough observations. If there is no prior knowledge available, so called *uninformative* priors are being used instead. *Uninformative* is again a misnomer – what is meant by these is the fact that they are not subjectively elicited. Fully Bayesian model is then a model in which *all* parameters have some prior assigned. An interesting property of Bayesian approach arise – random variables (including latent variables) and parameters are treated in the same way.

1.4 Statistical inference

Process of inferring properties of the data underlying distribution is called statistical inference[12]. Formally, it's justification of restricting parameter space based on the data (by e.g. choosing a point estimate for a given parameter).

Unfortunately, for many real world scenarios, the approach of directly inferring the properties is not possible. One of many reasons can be to high dimensionality of the task or the form of the posterior distribution may be to complex, let alone the fact that it does not have to have analytical solution[13]. In these scenarios, approximation methods are being used instead.

1.4.1 Approximation methods

Using a breakdown from [13], approximation methods can be divided to two groups based on their stochastic or deterministic behavior. A notable example of the former is Markov chain Monte Carlo[13, (Chap.11)]. A representative approach of the latter is e.g. variational inference described in detail in section Section 1.6. The main difference is the fact that stochastic schemes, *given infinite resources*, are guaranteed to get the exact results and the approximation arise from the finite amount of those resources. On contrary, deterministic approximation schemes are based on *analytical approximations* to the posterior distribution and hence are generally not able to generate exact results. Both approaches are then complementary to each other, one being useful for situation where the second is unsuitable and vice versa.

1.5 Conjugate distributions and exponential family

One of the remarkable properties of (1.1) is the fact that it can be used sequentially as an update based on new piece of data. If one performs the update and obtains $p(\mathbf{A}|\mathbf{B})$, this posterior can be used in the next iteration as a prior and so on. Using concept of conjugate priors exploits this remarkable property guaranteeing that one will get a tractable posterior after each iteration. In the most of the real world applications, using Bayes Theorem computationally means evaluating some integrals. For instance, to obtain the posterior, one must compute an integral $p(\mathbf{x}) = \int p(\mathbf{x}|\theta)p(\theta)d\theta$. To make use of the Bayes Theorem, one is then limited to problems where these integrals are tractable.

As already mentioned, this is done by using conjugate priors. Given a likelihood $p(\mathbf{x}|\theta)$, a family of prior distributions is chosen so that the integral $\int p(\mathbf{x}|\theta)p(\theta)d\theta$ is tractable and the result is in the same family as the prior. It is worth mentioning that these goals are usually in conflict. A tractable result would be obtained if we just used a constant as the distribution family, but the update would no necessarily result in a constant.

In practice, there is such a family, containing many of the most commonly used distributions such as Gaussian or Beta distribution. This family is called exponential family and guarantees

that if the model belongs to an exponential family and a conjugate prior distribution is selected, the resulting posterior of the update is analytically tractable.

1.5.1 Formal definition

A formal definition of the above can be found in [14] and is restated here since the forms from definitions are going to be used further along.

Definition 2. (Exponential family distribution). An exponential family distribution of a random variable y conditioned on x and with parameter Θ is a distribution whose probability density function can be written in the form

$$p(y|x,\Theta) = \exp\Big\{\eta^T T - A(\eta) + k(x,y)\Big\} \tag{1.2}$$

where $\eta = \eta(\Theta)$ is the natural parameter of the exponential family distribution, $T := T(x,y)$ is a sufficient statistic encompassing all information provided by x and y about the parameter Θ and

$$A(\eta) = \log \int \exp\Big\{\eta^T T + k(x,y)\Big\} \mathrm{d}x\mathrm{d}y \tag{1.3}$$

where the integral is over the space of x and $y$and $A(\eta)$ is a known log-partition function normalizing the density, and $k(x,y)$ is a known function independent of the parameter.

The form (1.2) is not unique since the product $\eta^T T$ can result in same the same number when each multiplied by appropriate constant.

Definition 3. (Conjugate prior distribution for Θ). Assume that a random variable y is conditioned by a variable x and obeys an exponential family distribution – model – with a parameter Θ. A prior distribution for Θ conjugate to the model is characterized by conjugate hyperparameters Ξ and Υ and has the probability density of the form

$$\pi(\Theta|\Xi,\Upsilon) = \exp\Big\{\eta^T \Xi + \Upsilon A(\eta) + l(\Xi,\Upsilon)\Big\} \tag{1.4}$$

where Ξ has the same size as $T, \Upsilon \in \mathbb{R}_+$ and $l(\Xi,\Upsilon)$ is a known function. The Ξ, Υ are called conjugate hyperparameters.

Substituting (1.2) and (1.4) into (1.1), the Bayesian update results into a simple summation

Theorem 4. *(Bayesian update under conjugacy). Assume that a random variable y conditioned by a variable x has an exponential family distribution with a parameter Θ, estimated by means of a conjugate prior distribution. The Bayesian update is then equivalent to the update of conjugate hyperparameters*

$$\Xi_t = \Xi_{t-1} + T_t \tag{1.5}$$
$$\Upsilon_t = \Upsilon_{t-1} + 1 \tag{1.6}$$

1.6 Variational Bayes inference

In this work, a family of deterministic approximation methods *variational inference (*or *variational Bayes)* is used, excellently described in [11, (p. 463)] and [15]. This technique is based on using solution of an optimization problem to statistical inference, more exactly finding an input which minimize a specific functional. As was described in 1.4.1, this method is guaranteed to get an exact result given some family of possible input functions over which one

minimizes. The approximation arise from limiting the possible inputs, for instance by considering only quadratic functions or, as is widely used and is also our case, functions which factorizes in a specific way.

Let $p(\mathbf{X}, \mathbf{Z})$ be a fully Bayesian model, where all prior distributions are given. Let denote the set of all parameters and all latent $\mathbf{Z} = \{\curvearrowright_1, \ldots, \curvearrowright_N\}$ and the set of all observed variables $\mathbf{X} = \{\curvearrowright_1, \ldots, \curvearrowright_N\}$. The goal of the inference is to find the posterior distribution $p(\mathbf{Z}|\mathbf{X})$ and the distribution of the model evidence $p(\mathbf{X}) = \int p(\mathbf{Z}, \mathbf{X})\ \mathrm{d}\mathbf{Z}$, which is usually unavailable. Unfortunately, in many real world scenarios, $p(\mathbf{Z}|\mathbf{X})$ is almost always intractable (by e.g. trying to integrate all configurations of the hidden variables in denominator). That is where *variational lower bound* comes in. Instead of trying to compute $p(\mathbf{Z}|\mathbf{X})$ directly, we consider $q(\mathbf{Z})$ which is as close approximation as possible to the former and has a convenient and tractable form (their expectations are computable). Furthermore, these approximate distributions can also have their own *variational parameters* which are considered to be in $\mathbf{Z}$ as well. As a measure of closeness between the approximate $q(\mathbf{Z})$ and $p(\mathbf{Z}|\mathbf{X})$, Kullback-Leibler (KL) divergence (1.7) is used[2].

$$\mathrm{KL}(q||p) = -\int q(\mathbf{Z}) \ln \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})}\ \mathrm{d}\mathbf{Z} \tag{1.7}$$

In a case when $p(\mathbf{X}|\mathbf{Z}) = q(\mathbf{Z})$, Kullback-Leibler divergence is effectively zero. The goal is the following optimization problem:

$$q(\mathbf{Z}) = \arg\min_{q(\mathbf{Z})} \mathrm{KL}(q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X})). \tag{1.8}$$

The optimization (1.8) cannot be performed directly. To compute $\mathrm{KL}(q||p)$, the unknown evidence $p(\mathbf{X})$ is needed, as can be seen from the following derivation

$$\mathrm{KL}(q||p(\mathbf{Z}|\mathbf{X})) = \mathbb{E}[\ln q] - \mathbb{E}[\ln p(\mathbf{Z}|\mathbf{X})] = \mathbb{E}[\ln q] - \mathbb{E}[\ln p(\mathbf{Z}, \mathbf{X})] + \ln p(\mathbf{X})$$

.

Instead of trying to compute $\mathrm{KL}(q||p)$, we are going to optimize an alternative objective that is equivalent to $\mathrm{KL}(q||p)$ up to an added constant. Let us then define *variational lower bound* (also called *ELBO*) as follows:

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})}\ \mathrm{d}\mathbf{Z} = \mathbb{E}[\ln p(\mathbf{Z}, \mathbf{X})] - \mathbb{E}[\ln q(\mathbf{Z})]. \tag{1.9}$$

It can be shown that$\mathcal{L}(q)$ has an important property being a lower bound of the log probability $\ln p(\mathbf{X}) \geq \mathcal{L}(q)$. Henceforth, when one wishes to maximize marginal $p(\mathbf{X})$, he can instead maximize its variational lower bound $\mathcal{L}(q)$.

Combining (1.9) and (1.7), the following relationship can be derived:

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \mathrm{KL}(q||p). \tag{1.10}$$

Since $p(\mathbf{X})$ doesn't depend on q, maximizing $\mathcal{L}(q)$ is possible and it is equivalent to minimizing $\mathrm{KL}(q||p)$.

Again, since we don't know the true posterior distribution, we work with some family of distributions $q(\mathbf{Z})$ for which $\mathcal{L}(q)$ becomes tractable and search for the candidate which maximizes it. Obviously, the choice of the distribution family is critical here – while it needs be tractable, it still needs to be flexible enough (as much as possible) to provide *accurate enough* approximation, goals usually going against each other. This is the part from where approximation arise in variational Bayes as has been discussed in 1.4.1.

[2]We omit arguments of distributions where possible for readability. Most often: $q := q(\mathbf{Z})$ and $p = p(\mathbf{Z}|\mathbf{X})$

1.6.1 Factorized distributions

A common choice for such a family of distributions described in the chapter Section 1.6 has been *factorized distributions* formalized in physics theory called *mean field theory* [16]. The assumption on the family is quite simple – we treat each variable in $\mathbf{Z}$ as independent, in other words, the family **factorizes**:

$$q(\mathbf{Z}) = \prod_{i=1}^{M} q_i(\mathbf{Z})$$

It is clear that the assumption is often not met. Variables are dependent in the real world – it is the reason why it is so hard to obtain the posterior distribution directly.

Notice that we do not require any specific form of $q_i(\mathbf{Z_i})$. Also, it is worth noting that this is *not* a model of the observed data – it is the ELBO and KL minimization problem, which connects this to the model and data.

1.7 Coordinate ascent mean-field variational

Coordinate ascent mean-field variational (*CAVI*) is a commonly used algorithm for solving optimization problem (1.8)[15][13].

1.7.1 Derivation

Optimal $q_i^*(\mathbf{Z_i})$

First, consider the complete conditional of $\mathbf{Z_i}$, which is $p(\mathbf{Z_i}|\mathbf{Z_{-i}}, \mathbf{X})$. If we fix all other variational factors $q_l(\mathbf{Z_l}), l \neq i$, the optimal $q_i(\mathbf{Z_i})$ satisfies the following

$$q_i^*(\mathbf{Z_i}) \propto \exp\{\mathbb{E}_{-i}[\ln p(\mathbf{Z_i}|\mathbf{Z_{-j}}, \mathbf{X})]\} \propto \exp\{\mathbb{E}_{-i}[\ln p(\mathbf{Z_i}, \mathbf{Z_{\{-i\}}}, \mathbf{X})]\} \tag{1.11}$$

where in the last term, we use the fact that $q_{-i}(\mathbf{Z_{-i}}) = \prod_{l \neq i} q_l(\mathbf{Z_l})$. By $\mathbb{E}_{-i}[...]$, we denote expectation over all q distributions *except* q_i. Similarly, $\mathbf{Z_{\{-i\}}}$ means all latent variables except $\mathbf{Z_i}$.

Obtaining KL

Let us now rewrite $\mathcal{L}(q)$ in (1.9) as a function of $q_i(\mathbf{Z_i})$

$$\mathcal{L}(q_i) = \mathbb{E}_i[\mathbb{E}_{-i}[\ln p(\mathbf{Z_i}, \mathbf{Z_{\{-i\}}}, \mathbf{X})]] - \mathbb{E}_i[\ln q_i(\mathbf{Z_i})] + C \tag{1.12}$$

where in the first term is just an iterated expectation and in the second we retain only product term containing $q_i(\mathbf{Z_i})$, while other parts can be moved to a constant C thanks to using factorized distribution.

Now it's easy to see that (1.12) is a negative KL divergence between $q_i(\mathbf{Z_i})$ and $q_i^*(\mathbf{Z_i})$. That gives as all necessary part for the CAVI algorithm as described in [15]:

Algorithm 5. *CAVI*
Input: *A model $p(\mathbf{X}, \mathbf{Z})$, a dataset $\mathbf{X}$*
Output: *A variational density $q(\mathbf{Z}) = \prod_{i=1}^{m} q_i(\mathbf{Z_i})$*
Initialize: *Variational factors $q_i(\mathbf{Z_i})$*
while *the $\mathcal{L}(q)$ has not converged*
  for *$i \in \{1, ..., m\}$* ***do***
    Set $q_i(\mathbf{Z_i}) \propto \exp\{\mathbb{E}_{-i}[\ln p(\mathbf{Z_i}, \mathbf{Z_{\{-i\}}}, \mathbf{X})]\}$

end
Compute $\mathcal{L}(q) = \mathbb{E}[\ln p(\mathbf{Z}, \mathbf{X})] - \mathbb{E}[\ln q(\mathbf{Z})]$
end
return $q(\mathbf{Z})$

It is guaranteed that this algorithm converge to a local minimum.

1.8 Example

Let us demonstrate the above with an example from [13].

The model Consider the i.i.d dataset $\mathbf{X} = \{x_1, \cdots, x_N\}$ generated by (unknown) Gaussian distribution $\mathbf{X} \sim \mathcal{N}(\mu, \tau^{-1})$. The goal is to infer posterior $p(\mu, \tau|\mathbf{X})$ and the joint probability is

$$p(\mathbf{X}, \mu, \tau) = p(\mathbf{X}|\mu, \tau)p(\mu|\tau)p(\tau)$$

Factorized distribution Let us now approximate $p(\mu, \tau|\mathbf{X})$ by $q(\mu, \tau)$ with the assumption that q factorizes: $q(\mu, \tau) = q(\mu)q(\tau)$.

Conjugated priors The complete data likelihood is Gaussian distribution and hence in exponential family. We set a non-informative conjugated priors for the μ and τ as follows

$$\begin{aligned}\mu &\sim \mathcal{N}(\mu_0, (\lambda_0\tau)^{-1}) \\ \tau &\sim \text{Gamma}(a_0, b_0)\end{aligned}$$

where the hyperparameters $\mu_0, \lambda_0, a_0, b_0$ are initially set to some small positive number.

Form of $q(\mu)$ and $q(\tau)$ This is usually the hardest part of variational inference. The derivation of the following terms is according to (1.11) and in detail can be found in [13].

The derivations yields following for $q^*_\mu(\mu)$

$$\begin{aligned}q^*_\mu(\mu) &\sim \mathcal{N}(\mu \mid \mu_N, \lambda_N^{-1}) \\ \mu_N &= \frac{\lambda_0\mu_0 + N\bar{x}}{\lambda_0 + N} \\ \lambda_N &= (\lambda_0 + N)\mathbb{E}_\tau[\tau] = (\lambda_0 + N)\frac{a_N}{b_N} \\ \bar{x} &= \frac{1}{N}\sum_{n=1}^{N} x_n\end{aligned}$$

and for $q^*_\tau(\tau)$:

$$
\begin{aligned}
q_\tau^*(\tau) &\sim \text{Gamma}(\tau \mid a_N, b_N) \\
a_N &= a_0 + \frac{N+1}{2} \\
b_N &= b_0 + \frac{1}{2}\mathbb{E}_\mu[\sum_{n=1}^{N}(x_n-\mu)^2 + \lambda_0(\mu-\mu_0)^2] = \\
&= b_0 + \frac{1}{2}\left[(\lambda_0+N)\left(\lambda_N^{-1}+\mu_N^2\right) - 2\left(\lambda_0\mu_0 + \sum_{n=1}^{N} x_n\right)\mu_N + \left(\sum_{n=1}^{N} x_n^2\right) + \lambda_0\mu_0^2\right]
\end{aligned}
$$

Perform updates based on CAVI Initially, compute $N, \sum_{n=1}^{N} x_n, \sum_{n=1}^{N} x_n^2$ and set $\lambda_{N,t}$ to some random value. Then iterate these steps until convergence:

1. plug in $\lambda_{N,t}$ to obtain $b_{N,t}$
2. compute new $\lambda_{N,t+1}$ based on $b_{N,t}$ and all other terms needed

Results After m iterations, we are now having new values of all parameters including hyper-parameters. Hence we are able to approximate the posterior distribution. The joint probability is

$$
\begin{aligned}
p(\mathbf{X}, \mu, \tau) &= p(\mathbf{X}|\mu,\tau)p(\mu|\tau)p(\tau) = \\
&= \prod_{n=1}^{N} \mathcal{N}(x|\mu_N, \tau_N^{-1})\mathcal{N}(\mu_N, (\lambda_N\tau_N)^{-1})\text{Gamma}(a_N, b_N)
\end{aligned}
$$

Chapter 2

Distributed Kalman filter

This chapter firstly describes basic (non-distributed) version of Kalman filter. This is later extended to a distributed variant.

2.1 Introduction

Kalman filter (KF) is widely used adaptive filtering method firstly described in *A New Approach to Linear Filtering and Prediction Problems* by R.E. Kalman in 1960. The filter is a recursive solution to the discrete-data linear filtering problem [17]. The filter is defined by a set of relatively simple equations and allows for efficient online computation without having to know the whole history of data and it is the optimal linear filter (under several conditions described below). It is one of the solution to one of the most fundamental problems in control theory, the linear quadratic Gaussian control problem. Over the years, numerous variants and extensions have been derived from the original Kalman filter and this thesis is concerned with one such extension – distributed Kalman filtering. See Section 2.6 for a brief overview of used methods.

Kalman filter has been thoroughly studied over the years and also adopted by multiple industries, having a vast amount of applications wherever an estimation of the state of process is needed and noise in prediction or observation is present. Because of its nature, it is commonly used in navigation and control of robots and vehicles[18], including spacecrafts[19]. It can be applied in time series analysis and hence is used in fields such as econometrics[20] and economics[21] or signal processing[22]. Various applications can be found in a medicine, such as neuroscience[23].

With respect to the current development of the so called Internet of Things and development of cheap measurements sensors and microcomputers, Distributed Kalman Filtering (DKF) is becoming even more relevant. As one of the solution to *sensor fusion* problem, it has been under an intensive research as can be seen in [24]. A network consisting of multiple sensors has many advantages over a single sensor scenario, such as robustness to noise, better field of view or, if designed correctly, not having a single point of failure [25]. Applications ranges from wireless networks[26], precision agriculture[**Adamchuk2011**], military and civil surveillance, medical applications, nuclear hazard assessment and others[14].

2.2 Description

The basic Kalman filter is used to estimate the current state of a linear dynamic system in case of noisy measurements. The state can be described by various variables, such as position and velocity. To use a Kalman Filter, one needs to specify system's:

1. control inputs, if any, such as sending a signal to steer a wheel
2. dynamic model, such as physical laws of motion
3. measurements, such as readings from sensors

The filter is useful in situations when dynamic model nor measurements cannot be entirely trusted (otherwise it one would not need to use it at all). It combines the information and while allowing for some noise – uncertainty – during prediction and measurements. The both pieces of information are weighted by so called Kalman's gain and depending on its value favors prediction over measurements and vice versa. The whole process is performed recursively, remembering the last estimate and covariance and hence does not need an entire history, which makes it attractive where memory is expensive. As a result a less noisy (hence a filter) and if used appropriately, better estimates are generated in each step, lying in between measurements and prediction.

Observable parameters coming from measurements does not have to be necessarily complete in a sense of describing the system in its entirety. It is a common case in which one observes only a subset of state variables and the filter can still estimate the entire state.

The distributed extension is easy to describe – instead of having only a single sensor, the system is measured by multiple of them and then somehow combining these measurements together. Despite its simple high level description, multisensor network of sensors must deal with various non-trivial problems and the whole computation is complicated based on the selected parameters of the network's behavior and topology. It also means getting into a distinctive research area of distributed data fusion[27] or consensus theory, which are out of the scope of this thesis.

obrázek:
https://medium.com/@mithi/object-tracking-and-fusing-sensor-measurements-using-the-extended-kalman-filter-algorithm-part-1-f2158ef1e4f0
https://www.google.cz/search?q=kalman+filter&safe=off&source=lnms&tbm=isch&sa=X&ved=0ahUKEv

2.3 Mathematical definition

Definition 6. Let $\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{u}_k \in \mathbb{R}^n$ be the true state vectors in times $k, k-1$ and control vector, respectively. $\mathbf{z}_k \in \mathbb{R}^m$ is the measurement vector, $\mathbf{F}_k, \mathbf{B}_k \in \mathbb{R}^{n \times m}$ the state transition matrix and control matrix, respectively and $\mathbf{H}_k \in \mathbb{R}^{m \times n}$ is the observation matrix. $\mathbf{w}_k \in \mathbb{R}^n$ and $\mathbf{v}_k \in \mathbb{R}^m$ are process noise and observation noise, respectively. The filter assumes that the state evolves from (discrete) time $k-1$ to a new state in time k according to:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \tag{2.1}$$

and then an (imperfect) observation is made

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{2.2}$$

In this work, the $\mathbf{B}_k \mathbf{u}_k$ part is omitted as is common in many scenarios, where control input does not make sense (such as temperature sensors). The process and observation noise $\mathbf{w}_k, \mathbf{v}_k$ are assumed to be zero mean (multivariate in case of $\mathbf{w}_k$) Gaussian white noise with covariances $\mathbf{Q}_k, \mathbf{R}_k$:

$$\begin{aligned} \mathbf{w}_k &\sim \mathcal{N}(0, \mathbf{Q}_k) \\ \mathbf{v}_k &\sim \mathcal{N}(0, \mathbf{R}_k) \end{aligned}$$

The update is then performed according to:

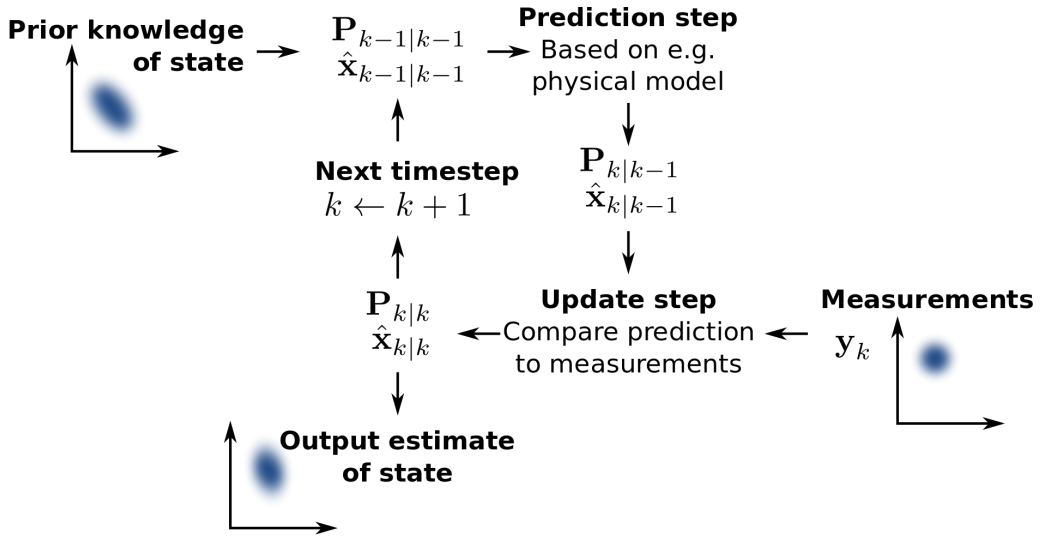


Figure 2.1: The whole process of Kalman filter. We start with an information from the previous step $(k-1|k-1)$, then perform prediction to get expected values $(k|k-1)$, after which we incorporate measurements information and perform an update to get the corrected state $(k|k)$. Source: https://www.wikiwand.com/en/Kalman_filter

Start with basic KF, then extension to (distributed) Diffusion Kalman Filtering. Nice intro with pictures in [25], simple def in [24] (eq. 2.1) linking to https://link.springer.com/article/10.1007/s11424-012-0275-2,
Additional definitions in [28] or (more complex description) in [29].

2.4 Note on optimality

The Kalman filter is the optimal filter if and only if the following conditions are fulfilled, which is rarely the case:

1. the model is exactly reflecting the reality
2. the noise is white
3. the covariances of the noise are exactly known

Optimality for distributed case: Track-to-Track Fusion scheme, e.g. in [25] in Introduction page 3.

why it's the problem, hence the next section ->

2.5 Noise analysis

What is their purpose and what problems we have with them, why it's necessary to nicely estimate them otherwise everything breaks down (described in the main VBAKF article on the beginning) + optimality issue.

2.6 Methods

This is what I want: [24]
Survey of currently used methods to deal with unknown noise

2.7 Distributed Kalman Filter

Chapter 3

Distributed Variational Bayes Adaptive Kalman Filtration

This chapter describes main contributions of this work. It is a derivation of an improvement of a Variation Bayes Adaptive Kalman Filtration (VBAKF) described [30] by partially sharing parameters in diffusion networks from [14]. Up to the author knowledge, there has not been a similar attempt to combine these two, although [14] mention these as a future possibility.

We also propose a simplified version of the algorithm from [30] using exponential family form, as we believe that the resulting update equations are more elegant, concise and understandable.

3.1 Problem formulation

As has been described in

tady by měl být odkaz do teorie kde bude to co oni ají v Main Results.A + problémy s estimation šumu a proč se to musí řešit elegantně.

, our aim is ti describe a model in which we will infer not only $\mathbf{x}_k$, but also PNCM $\mathbf{P}_{k|k-1}$ and MCNM $\mathbf{R}_k$ as opposed to traditional Kalman filter where these two are considered static

Popsat co ty zkratky znamenají

.

Based on explanation in

Odkaz, jak se ty rovnice přeloží přes HMM do pravděpodobnosti:
https://www.wikiwand.com/en/Kalman_filter#/Relationship_to_recursive_Bayesian_estimation

, the predicted PDF for $\mathbf{x}_k$ and likelihood PDF

what is PDF

follows Gaussian distribution:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}, \mathbf{P}_{k|k-1}) = \mathrm{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \tag{3.1}$$

$$p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{R}_k) = \mathrm{N}(\mathbf{z}_k; \mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k) \tag{3.2}$$

where

- $\mathrm{N}(\cdot; \mu, \Sigma)$ is Gaussian PDF with corresponding mean μ and covariance matrix Σ
- $\hat{\mathbf{x}}_{k|k-1}$ is the predicted state vector based on the model in

  asi zase 2

- $\mathbf{P}_{k|k-1}$ is the predicted PNCM based on the model in

asi zase 2

We emphasize that $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$ are both inaccurate because precise $\mathbf{Q}_k$ from the underlying update equation in

to samé co v předchozím todo

is unknown and we would not have to do the whole machinery if it was known.

3.1.1 Prior distributions for $\mathbf{P}_{k|k-1}$ and $\mathbf{R}_k$

We are going to leverage the benefits of good choice of priors to make the whole solution tractable outlined in Section 1.5. To ensure that the posterior belongs to the exponential family, we choose Inverse-Wishart distribution which is in a Bayesian statistic a common choice as a conjugate prior for the covariance matrix of a multivariate normal distribution with known mean[31]. That is the case for $\mathbf{P}_{k|k-1}$ and $\mathbf{R}_k$, which are both the covariances matrices of Gaussian PDFs as is in equations (3.1) and (3.2).

Definition 7. Inverse-Wishart distribution is a probability distribution on real-value positive-definite matrices. We say that $\mathbf{X}$ follows the Inverse-Wishart distribution with ν degrees of freedom and scale matrix Ψ, i.e. $\mathbf{X} \sim \mathrm{IW}(\Psi, \nu)$, if $\mathbf{X}^{-1}$ has a Wishart distribution $\mathbf{X} \sim \mathrm{W}(\Psi^{-1}, \nu)$. The density is

$$\frac{|\boldsymbol{\Psi}|^{\frac{\nu}{2}}}{2^{\frac{\nu p}{2}}\Gamma_p(\frac{\nu}{2})}\,|\mathbf{X}|^{-\frac{\nu+p+1}{2}}\,e^{-\frac{1}{2}\mathrm{tr}(\boldsymbol{\Psi}\mathbf{X}^{-1})} \tag{3.3}$$

where $\boldsymbol{\Psi} > \mathbf{0} \in \mathbb{R}^{p\times p}$ is a positive-definite scale matrix, $\nu > p-1, \nu \in \mathbb{R}$ are degrees of freedom, $\Gamma_p(\cdot)$ is the multivariate gamma function and tr is the trace function. The exponential family form we are going to take advantage of in the following text is

$$\exp\left\{\begin{bmatrix}\mathbf{X}^{-1}\\ \ln|\mathbf{X}|\end{bmatrix}^{\!\top}\begin{bmatrix}-\frac{1}{2}\boldsymbol{\Psi} & -\frac{\nu+p+1}{2}\end{bmatrix} - \left(\frac{\nu}{2}(p\ln 2 - \ln|\boldsymbol{\Psi}|) + \ln\Gamma_p\left(\frac{\nu}{2}\right)\right)\right\} \tag{3.4}$$

and the corresponding parts of the form from Section 1.5 are

$$\begin{aligned}\eta(\mathbf{X}) &= \begin{bmatrix}\mathbf{X}^{-1}\\ \ln|\mathbf{X}|\end{bmatrix}\\ \Xi(\Psi,\nu,p) &= \begin{bmatrix}-\frac{1}{2}\boldsymbol{\Psi} & -\frac{\nu+p+1}{2}\end{bmatrix}\\ A(\eta) &= \frac{\nu}{2}(p\ln 2 - \ln|\boldsymbol{\Psi}|) - \ln\Gamma_p\left(\frac{\nu}{2}\right)\end{aligned}$$

The first element Ξ_1 of a *pseudovector* Ξ is a matrix, why the second Ξ_2 is a scalar. The advantages of this notation will be clearer when an addition to natural parameters is carried over. The expected value of the IW distributed variable is

$$\mathrm{E}[\mathbf{X}] = \frac{\Psi}{\nu-p-1} \tag{3.5}$$

$$= \frac{2}{2\Xi_2 + p + 1}\Xi_1 \tag{3.6}$$

Following from the above, the probabilities for $\mathbf{P}_{k|k-1}$ and $\mathbf{R}_k$ are hence

$$p(\mathbf{P}_{k|k-1}|\mathbf{z}_{1:k-1}) = \text{IW}(\mathbf{P}_{k|k-1}; \hat{t}_{k|k-1}, \hat{\mathbf{T}}_{k|k-1}) \tag{3.7}$$

$$p(\mathbf{R}_k|\mathbf{z}_{1:k-1}) = \text{IW}(\mathbf{R}_k; \hat{u}_{k|k-1}, \hat{\mathbf{U}}_{k|k-1}) \tag{3.8}$$

Next, prior parameters for $\hat{t}_{k|k-1}, \hat{\mathbf{T}}_{k|k-1}, \hat{u}_{k|k-1}, \hat{\mathbf{U}}_{k|k-1}$ to represent the initial knowledge as precisely as possible. These parameters will be consumed later into prior's hyperparameters in Ξ.

Hyperparameters for PNCM

The model from [30] proposes the following initial parameter setting for the prior information for $\mathbf{P}_{k|k-1}$

$$\frac{\hat{\mathbf{T}}_{k|k-1}}{\hat{t}_{k|k-1} - n - 1} = \tilde{\mathbf{P}}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \tilde{\mathbf{Q}}_{k-1} \tag{3.9}$$

$$\hat{\mathbf{T}}_{k|k-1} = \tau\tilde{\mathbf{P}}_{k|k-1} \tag{3.10}$$

$$\hat{t}_{k|k-1} = n + \tau + 1 \tag{3.11}$$

where $\tilde{\mathbf{P}}_{k|k-1}, \tilde{\mathbf{Q}}_{k-1}$ are the *nominal* PECM and PNCM, respectively. The latter, together with $\tau \geq 0$, are model's parameters and their choice is discussed later. Substituting $\mathbf{P}_{k|k-1}, \hat{t}_{k|k-1}, \hat{\mathbf{T}}_{k|k-1}$ into (3.4) and combining with (3.10) and (3.11), we get

$$\eta(\mathbf{P}_{k|k-1}) = \begin{bmatrix} \mathbf{P}_{k|k-1}^{-1} \\ \ln|\mathbf{P}_{k|k-1}| \end{bmatrix} \tag{3.12}$$

$$\Xi(\hat{\mathbf{T}}_{k|k-1}, \hat{t}_{k|k-1}, n) = \begin{bmatrix} -\frac{1}{2}\hat{\mathbf{T}}_{k|k-1} & -\dfrac{\hat{t}_{k|k-1} + n + 1}{2} \end{bmatrix} = \tag{3.13}$$

$$= \begin{bmatrix} -\frac{1}{2}\tau\tilde{\mathbf{P}}_{k|k-1} & -\frac{1}{2}\tau - 1 - n \end{bmatrix} \tag{3.14}$$

where n is the dimension of $\hat{\mathbf{T}}_{k|k-1}$.

Hyperparameters for MCNM

According to (1.1), the prior for $p(\mathbf{R}_k|\mathbf{z}_{1:k-1})$ is

$$p(\mathbf{R}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{R}_k|\mathbf{R}_{k-1})p(\mathbf{R}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{R}_{k-1} \tag{3.15}$$

To make sure that $p(\mathbf{R}_k|\mathbf{z}_{1:k-1})$ is also Inverse-Wishart distributed, the $p(\mathbf{R}_k|\mathbf{R}_{k-1})$ must have a suitable form so the Bayesian update results in the exponential family. Allowing slowly varying MNCM, as is the case in many real world applications, a simple heuristic from [32] is chosen to deal with this uncertainty. That is spreading previous posteriors by a factor of ρ, which is then a model parameter

$$\hat{u}_{k|k-1} = \rho(\hat{u}_{k-1|k-1} - m - 1) + m + 1 \tag{3.16}$$

$$\hat{\mathbf{U}}_{k|k-1} = \rho\hat{\mathbf{U}}_{k-1|k-1} \tag{3.17}$$

where m is the size of matrix $\hat{\mathbf{U}}_{\cdot|\cdot}$. The initial MNCM $\mathbf{R}_0$ is also considered to be Inverse-Wishart distributed

$$p(\mathbf{R}_0 = \mathrm{IW}(\mathbf{R}_0; \hat{u}_{0|0}, \hat{\mathbf{U}}_{0|0}) \tag{3.18}$$

Similarly to the PNCM case in (3.9), the initial prior information for the mean value of $\mathbf{R}_0$ is set as the initial nominal MNCM $\tilde{\mathbf{R}}_0$

$$\frac{\hat{\mathbf{U}}_{0|0}}{\hat{u}_{0|0} - m - 1} = \tilde{\mathbf{R}}_0 \tag{3.19}$$

where $\tilde{\mathbf{R}}_0$ is then a model parameter.
Substituting $\mathbf{R}_{k-1}, \hat{u}_k, \hat{\mathbf{U}}_k$ into (3.4) and combining with (3.17) and (3.16), we get

$$\eta(\mathbf{R}_{k-1}) = \begin{bmatrix} \mathbf{R}_{k-1}^{-1} \\ \ln|\mathbf{R}_{k-1}| \end{bmatrix} \tag{3.20}$$

$$\Omega(\hat{\mathbf{U}}_k, \hat{u}_k, m) = \begin{bmatrix} -\frac{1}{2}\hat{\mathbf{U}}_k & -\dfrac{\hat{u}_k + m + 1}{2} \end{bmatrix} = \tag{3.21}$$

$$= \begin{bmatrix} -\frac{1}{2}\rho\hat{\mathbf{U}}_k & -\frac{1}{2}\rho(\hat{u}_k - m - 1) - m - 1 \end{bmatrix} \tag{3.22}$$

where n is the dimension of $\hat{\mathbf{U}}_k$.

Choice of the nominal PNCM and MNCM

In the detailed analysis [30] can be found that to guarantee model's convergence, the initial nominal PNCM $\tilde{\mathbf{Q}}_k$ needs to be near the initial true PNCM $\mathbf{Q}_k$. In the paper and as well in this work, we set $\mathbf{Q}_k = \mathrm{diag}[\alpha_{1,k}, \ldots, \alpha_{n,k}], \alpha_{i,k} > 0$ In reality, this is done based on the state-of-the-art engineering knowledge in the domain. The same is applied for the initial nominal MNCM $\tilde{\mathbf{R}}_0$ and the choice is $\mathbf{R}_k = \mathrm{diag}[\beta_1, \ldots, \beta_m], \beta_i > 0$ and the reader is kindly redirected to the original study for more information.

3.2 Variational Bayes derivation

In order to to infer all $\mathbf{x}_k, \mathbf{P}_{k|k-1}$ and $\mathbf{R}_k$, a posterior PDF $p(\mathbf{x}_k, \mathbf{P}_{k|k-1}, \mathbf{R}_k)$ needs to be estimated. As thoroughly described in chapter Section 1.6, the Variation Bayes in our case expects the following approximation

$$p(\mathbf{x}_k, \mathbf{P}_{k|k-1}, \mathbf{R}_k|\mathbf{z}_{1:k}) \approx q(\mathbf{x}_k)q(\mathbf{P}_{k|k-1})q(\mathbf{R}_k) \tag{3.23}$$

where $q(\mathbf{x}_k), q(\mathbf{P}_{k|k-1}), q(\mathbf{R}_k)$ are factors of our factorized distribution $q(\cdot)$ by which we approximate the $p(\cdot)$. These factors are obtained using Variational Bayes machinery including minimizing the Kullback-Leibler divergence and using CAVI.

3.2.1 Probabilistic model

Combining (3.1), (3.2), (3.7) and (3.8), the join PDF can be factored as

$$p(\boldsymbol{\Xi}, \mathbf{z}_{1:k}) = p\big(\mathbf{z}_k|\mathbf{x}_k, \mathbf{R}_k\big)p\big(\mathbf{x}_k|\mathbf{z}_{1:k-1}, \mathbf{P}_{k|k-1}\big)p(\mathbf{P}_{k|k-1}|\mathbf{z}_{1:k-1})p(\mathbf{R}_k|\mathbf{z}_{1:k-1})p(\mathbf{z}_{1:k-1}) \tag{3.24}$$

and substituting appropriate distribution for these PDFs:

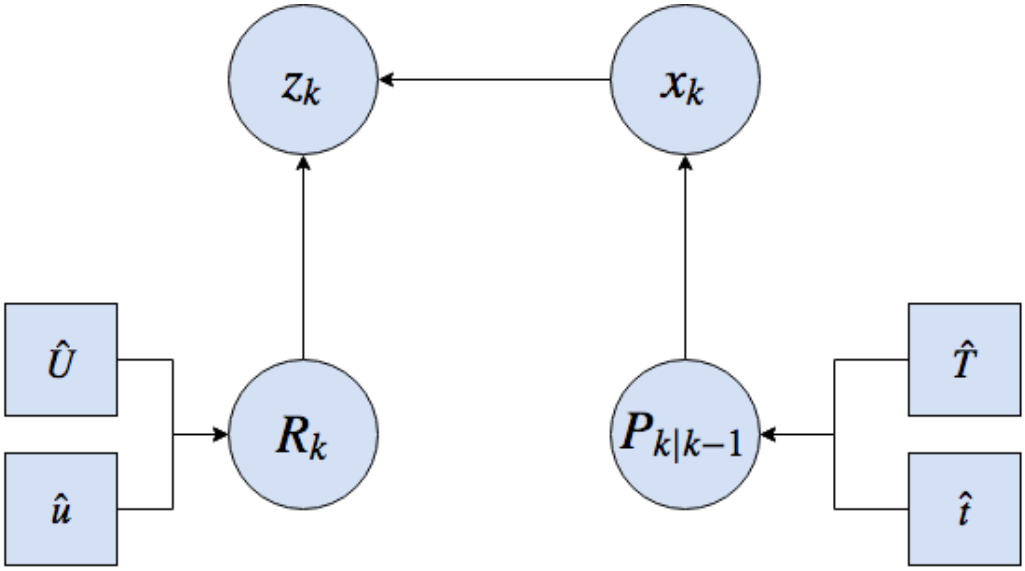


Figure 3.1: Graphical model of probabilistic model (3.24)

$$p(\Xi, \mathbf{z}_{1:k}) = \mathrm{N}(\mathbf{z}_k; \mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k)\mathrm{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \tag{3.25}$$

$$\times \mathrm{IW}(\mathbf{P}_{k|k-1}; \hat{t}_{k|k-1}, \hat{\mathbf{T}}_{k|k-1})\mathrm{IW}(\mathbf{R}_k; \hat{u}_{k|k-1}, \hat{\mathbf{U}}_{k|k-1}) \tag{3.26}$$

$$\times p(\mathbf{z}_{1:k-1}). \tag{3.27}$$

3.2.2 Deriving update equations

As opposed to [30], we are going to leverage advantages of exponential family forms to infer the update equations. Considering the exponential form of a Gaussian distribution

$$\mathrm{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \exp\left\{ \begin{bmatrix} \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ -\frac{1}{2}\boldsymbol{\Sigma}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{x} & \mathbf{x}\mathbf{x}^T \end{bmatrix} - (\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{1}{2}\ln|\boldsymbol{\Sigma}|) \right\} \tag{3.28}$$

$$= \exp\left\{ \begin{bmatrix} \boldsymbol{\Sigma}^{-1} \\ \ln|\boldsymbol{\Sigma}^{-1}| \end{bmatrix} \begin{bmatrix} -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T & -\frac{1}{2} \end{bmatrix} + \ln(2\pi) \right\} \tag{3.29}$$

where $\mathbf{x}\mathbf{x}^{\mathbf{T}}$ is an outer product, hence resulting into a matrix from $\mathbb{R}^{n\times n}$. Substituting appropriate factors for both $\mathbf{x}_k$ as well as $\mathbf{z}_k$ we get

$$\mathrm{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) = \exp\left\{ \begin{bmatrix} \mathbf{P}^{-1}_{k|k-1} \\ \ln|\mathbf{P}^{-1}_{k|k-1}| \end{bmatrix} \begin{bmatrix} -\frac{1}{2}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^T & -\frac{1}{2} \end{bmatrix} - k_1 \right\} \tag{3.30}$$

$$\mathrm{N}(\mathbf{z}_k; \mathbf{x}_k, \mathbf{R}_k) = \exp\left\{ \begin{bmatrix} \mathbf{R}_k^{-1} \\ \ln|\mathbf{R}_k^{-1}| \end{bmatrix} \begin{bmatrix} -\frac{1}{2}(\mathbf{z}_k - \mathbf{x}_k)(\mathbf{z}_k - \mathbf{x}_k)^T & -\frac{1}{2} \end{bmatrix} - k_2 \right\} \tag{3.31}$$

As can be seen, the natural parameters matches for (3.30) and (3.12) as well as for (3.31) and (3.20), which is of course intentional and result of clever choice of conjugated priors.

With respect to (1.5), the single step Bayesian update of Inverse-Wishart hyperparameters with Multivariate Normal Distribution can be written as

$$\Xi_i \leftarrow \underbrace{\begin{bmatrix} -\frac{1}{2}\boldsymbol{\Psi} \\ -\frac{\nu+p+1}{2} \end{bmatrix}}_{\Xi_{i-1}} + \underbrace{\begin{bmatrix} -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T \\ -\frac{1}{2} \end{bmatrix}}_{T(x,\mu)} \tag{3.32}$$

Utilizing the fact that

$$\mathrm{E}^i[-\frac{1}{2}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^T] = \mathbf{P}^i_{k|k} + (\hat{\mathbf{x}}^i_{k|k} - \hat{\mathbf{x}}_{k|k-1})(\hat{\mathbf{x}}^i_{k|k} - \hat{\mathbf{x}}_{k|k-1})^T \tag{3.33}$$

$$\mathrm{E}^i[-\frac{1}{2}(\mathbf{z}_k - \mathbf{H}_k\mathbf{x}_k)(\mathbf{z}_k - \mathbf{H}_k\mathbf{x}_k)^T] = \mathbf{H}_k\mathbf{P}^i_{k|k}\mathbf{H}_k^T + (\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}^i_{k|k})(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}^i_{k|k})^T \tag{3.34}$$

the single step i of the variational updates are

$$\Xi^n_{i+1} \leftarrow \Xi^{n-1} + \begin{bmatrix} -\frac{1}{2}(\mathbf{P}^i_{k|k} + (\hat{\mathbf{x}}^i_{k|k} - \hat{\mathbf{x}}_{k|k-1})(\hat{\mathbf{x}}^i_{k|k} - \hat{\mathbf{x}}_{k|k-1})^T) \\ -\frac{1}{2} \end{bmatrix} \tag{3.35}$$

$$\Omega^n_{i+1} \leftarrow \Omega^{n-1} + \begin{bmatrix} \mathbf{H}_k\mathbf{P}^i_{k|k}\mathbf{H}_k^T + (\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}^i_{k|k})(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}^i_{k|k})^T \\ -\frac{1}{2} \end{bmatrix} \tag{3.36}$$

where Ξ^{n-1}, Ω^{n-1} are hyperparameters obtained from the results of Kalman filter of the previous observation, derived using formulas (3.14) and (3.22).

Kalman correction

The last bit of the single CAVI iteration is the Kalman correction, which is the standard

$$\mathbf{K}_k^{(i+1)} = \hat{\mathbf{P}}^{(i+1)}_{k|k-1}\mathbf{H}_k^T\big(\mathbf{H}_k\hat{\mathbf{P}}^{(i+1)}_{k|k-1}\mathbf{H}_k^T + \hat{\mathbf{R}}_k^{(i+1)}\big)^{-1} \tag{3.37}$$

$$\hat{\mathbf{x}}^{(i+1)}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k^{(i+1)}\big(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}\big) \tag{3.38}$$

$$\mathbf{P}^{(i+1)}_{k|k} = \hat{\mathbf{P}}^{(i+1)}_{k|k-1} - \mathbf{K}_k^{(i+1)}\mathbf{H}_k\hat{\mathbf{P}}^{(i+1)}_{k|k-1} \tag{3.39}$$

3.3 Parameter diffusion

Thanks to using the exponentially family forms, parameter diffusion can be easily incorporated into the final algorithm – the nodes only need to exchange the hyperparameters Ξ^n and Ω^n. While the previous chapters were mostly based on algorithm described in [30], the parameter diffusion part is mostly based on [14]. We just present here the result algorithm, but the reader is advised to reach the paper for more details.

Consider a network of connected nodes, each of them doing their own observation. A single node i obtain a measurement in time t and proceed with Kalman filtration as above. After obtaining final $\mathbf{P}^{i,t}_{k|k}, \hat{\mathbf{x}}^{i,t}_{k|k}$ and also result hyperparameters $\Xi^{i,t}, \Omega^{i,t}$. The node then *contacts* its neighbor nodes $1, ..., d$ and attempts to gather all available hyperparameters $\Xi^{j,t}, \Omega^{j,t}, j \in \{1, ..., d\}$ based on the current conditions (e.g. only a single hop distance, or more). The node then performs a parameter fusion of these hyperparameters based on a chosen strategy represented by some

function $\Xi_*^{i,t} = g(\Xi^{i,t}, \Xi^{1,t}, ..., \Xi^{d,t})$. An example of such function may be just a simple averaging over all hyperparameters $g(\Xi^{i,t}, \Xi^{1,t}, ..., \Xi^{d,t}) = \frac{\sum_{s=\{i,1,...,d\}} \Xi^{s,t}}{d+1}$. The situation is for the hyperparameter $\Omega^{i,t}$ is analogical. The node then sends this updated hyperparameters $\Xi_*^{i,t}, \Omega_*^{i,t}$ back to the cooperating nodes, which incorporates this hyperparameter into their knowledge.

3.4 Algorithm summary

Here we summarize the previous pages into a single algorithm. The single step of the proposed algorithm for a single node is then

Algorithm 8. *1.* ***Inputs:*** $\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}, \hat{u}_{k-1|k-1}, \hat{\mathbf{U}}_{k-1|k-1}, \mathbf{F}_{k-1}, \mathbf{H}_k, \mathbf{z}_k, \tilde{\mathbf{Q}}_{k-1}, m, n, \tau, \rho, N$

2. Time Update of the Kalman Filter

jen odkaz jako jinde

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1}$$
$$\tilde{\mathbf{P}}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \tilde{\mathbf{Q}}_{k-1}$$

3. Initialize Ξ_{init} *by* (3.14), Ω_{init} *by* (3.22), $\hat{\mathbf{x}}_{k|k}^{(0)} = \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k}^{(0)} = \tilde{\mathbf{P}}_k$

For *i* ***in*** *N* ***do:***

4. Get new Ω^i *based on the* (3.36) *and compute the* $\mathrm{E}^i[\mathbf{R}_k]$ (3.6)

5. Get new Ξ^i *based on the* (3.35) *and compute* $\mathrm{E}^i[\mathbf{P}_{k|k-1}]$ *using* (3.6)

6. Perform Kalman correction (3.37)-(3.39) *using expected values to obtain corrected* $\hat{\mathbf{x}}_{k|k}^{i+1}, \mathbf{P}_{k|k-1}^{i+1}$

end for

7. Perform parameter diffusion, i.e. if hyperparameters from nodes $j \in \{1, ..., d\}$ *are available, get*

$$\Xi_*^{i,t} = g(\Xi^{i,t}, \Xi^{1,t}, ..., \Xi^{d,t}) \tag{3.40}$$
$$\Omega_*^{i,t} = g(\Omega^{i,t}, \Omega^{1,t}, ..., \Omega^{d,t}) \tag{3.41}$$

8. ***Returns:*** $\hat{\mathbf{x}}_{k|k}^{i+1}, \mathbf{P}_{k|k-1}^{i+1}, \Xi_*^{i,t}, \Omega_*^{i,t}$

Chapter 4

Results and simulations

The following chapter offers an in-depth analysis of a performance of the proposed algorithm. To guarantee a reproducible research, implementation details are mentioned to some degree and the code with the comparison is publicly available on `https://github.com/hnykda/kfsims`. This thesis focused mostly on state prediction accuracy and less on the noise analysis, although it is mentioned when it helps to understand the mechanism behind the performance.

Please note that some figures should be compared against each other as a trend, rather then individual lines. This is true especially in the case of the multi-node variants, when the author's goal is to demonstrate that all the nodes are having some specific property which would be hard to demonstrate by plotting only a single node.

4.1 Implementation

The implementation used in this work has been written in feature rich Python language ecosystem. Next to the CPython interpreter 3.6, following libraries (all licensed under MIT license) has been reused:

- scipy 1.0.0 – generator of random numbers from various distributions
- numpy 1.14.2 – arrays and mathematical operations, linear algebra
- networkx 2.1 – to create a network topology of measurement nodes
- jupyter 5.2.3 – "notebook" ecosystem for a rapid analysis
- filterpy 1.2.1 – reference implementation of a classic Kalman Filter
- matplotlib 2.2.2 – to create charts (including this thesis)

All the code is available on GitHub page `https://github.com/hnykda/kfsims` including the instruction how to replicate the analysis and testing environment. Most of the functionality has been written from scratch by the author while having a reference implementation by the supervisor (plus the model of the testing trajectory). The code is structured into several parts, notably objects `MeasurementNode` and `IWPrior` plus many of utility functions in the appropriate modules such as `exponential_family.py` or `network.py`, which are self-explanatory.

Comparison

The notation for the components of the state vector is kept as x_i in the charts. From the nature of the problem under study, it is clear that first two components which are observable

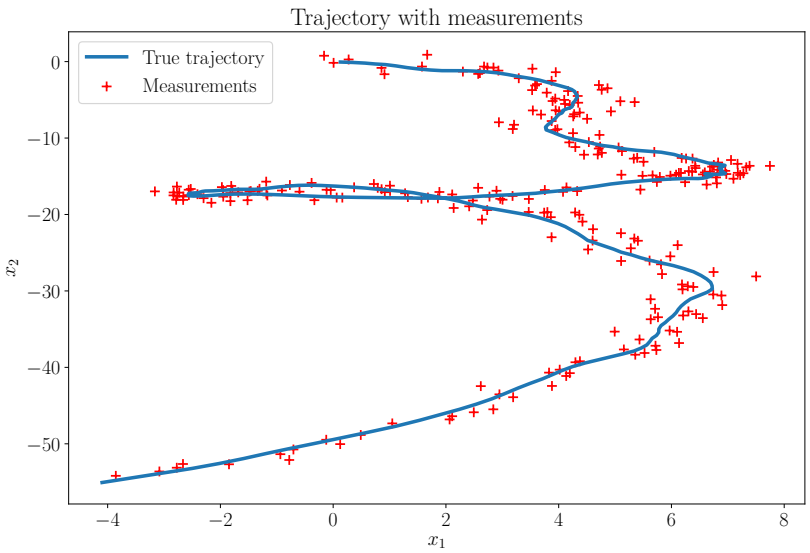


Figure 4.1: First and second component in the first row are observable, while the third and fourth is not.

behave in the similar way and the same is true for the latter two (which are not observable). To keep the analysis clean and concise, only a representative component is plotted if the behavior is significantly different.

The main metric used for comparison is root mean square error. Where appropriate, standard deviation of errors is listed. The result numbers discussed further are results of many iterations (more than 20) with different initial noise assigned to different nodes. It was although made sure that when comparing two different variants, same noise has been attributed to the appropriate elements. Finally, in situations when we are going to *visualize* matrices behavior, in our case matrices close to a matrix $c\mathrm{I}, c > 0$, a Frobenius matrix norm is used.

4.2 Testing data

Here we describe the dynamical system model in the standard notation suitable for the Kalman Filter. The state transition matrix was of the form

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{Q} = 2 \cdot \begin{bmatrix} t^3/3 & 0 & t^2/2 & 0 \\ 0 & t^3/3 & 0 & t^2/2 \\ t^2/2 & 0 & t & 0 \\ 0 & t^2/2 & 0 & t \end{bmatrix}, t = 0.1$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \mathbf{R} = 0.25 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

A random trajectory was then simulated for all four components using this model with multivariate Gaussian random noise added to the each "true" datapoint (with covariance $\mathbf{Q}$) as well as to the observations (with covariance $\mathbf{R}$). It can be though about as a point moving on a 2D plane. A typical result of the first two components of the test trajectory can be seen on Figure 4.1 on page 28 while the other two on Figure 4.2 on page 29. We always generated 250 time steps.

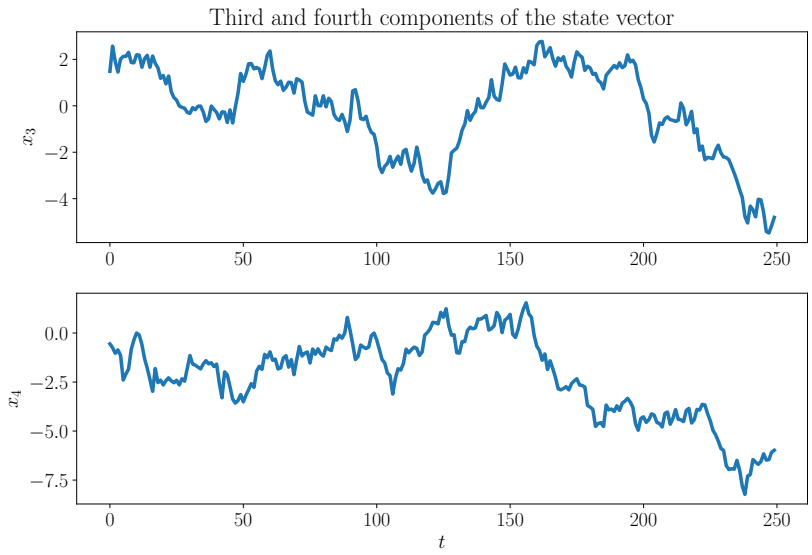


Figure 4.2: True third and fourth component of the state vector.

4.2.1 Noise

Different type of noises were tested to thoroughly examine the behavior of different filter under different circumstances. Examples for a static noise and variable noise can be found in Figure 4.3 on page 30 and Figure 4.4 on page 30. Even though only a single line is plotted, in the simulation both observable components distorted by the type of noise under study. Henceforth, for the sake of simplicity, MNCM was the factor we considered that can be slowly varying. In cases when the true covariance is unknown, the initial covariance was set to a base trajectory covariance $\mathbf{R}$, simulating an estimate one could obtain in the real application by other means.

It worth mentioning that because of the task's triviality, the CAVI algorithm usually converges in few iterations – less then 5. Therefore, number of iterations were set to conservative 10. There is an exhaustive analysis of algorithm stability and parameters effect in [30], where the reader is kindly referred to if she wishes to know more.

4.3 Single node VBAKF

In this section, we compare the VBAKF algorithm performance against the classic Kalman Filter (CKF). It is important to distinguish between the situation when the true noise covariance is known and when it is not. Another orthogonal property of the system is the variance of the noise covariance over time. Since the motivation of developing VBAKF is specially for the situations when the covariance is unknown or there is a risk of choosing a wrong one, it is clear where the VBAKF is expected to perform better.

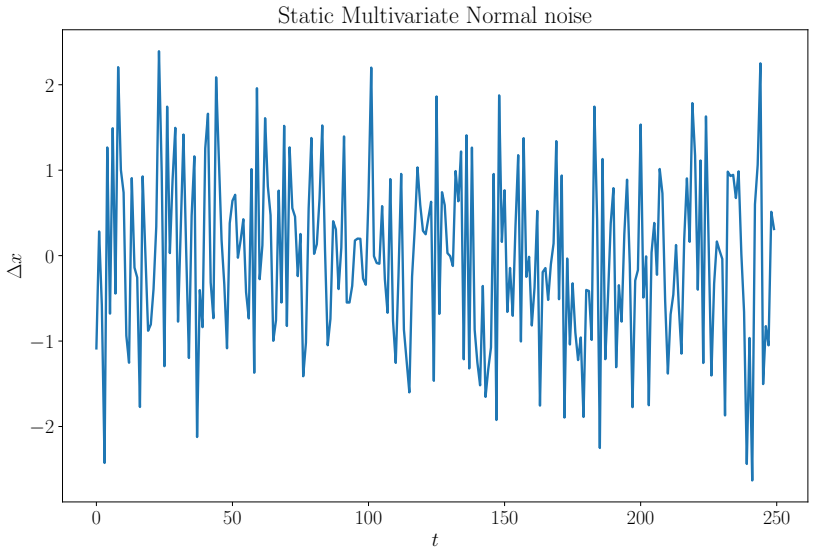


Figure 4.3: Multivariate Gaussian Random noise without variable variance.

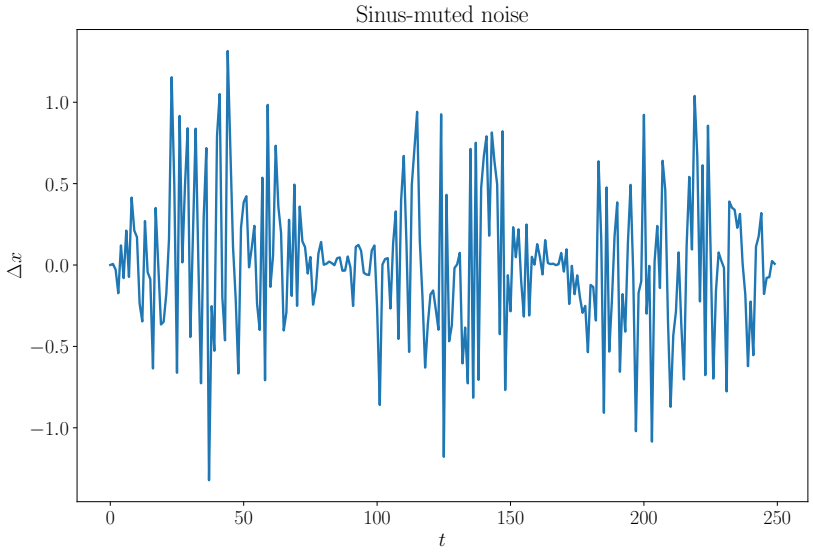


Figure 4.4: A variable variance in the noise is simulated in this example.

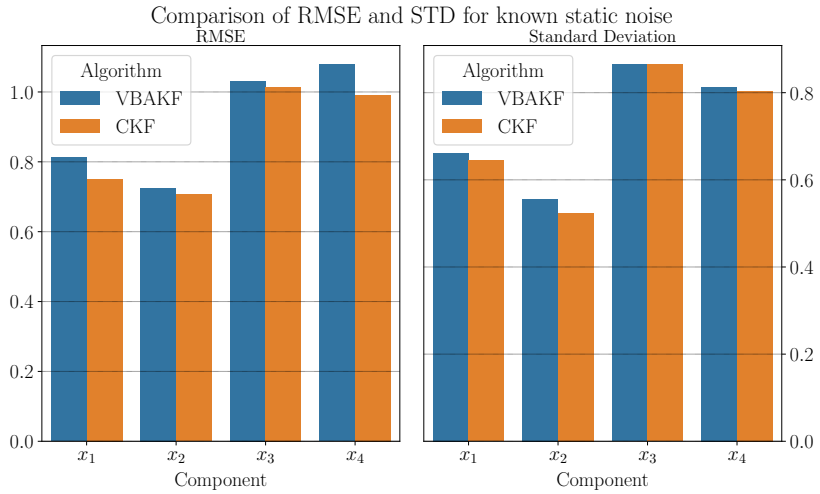


Figure 4.5: As expected, there is no performance gain in prediction(right) nor standard deviation (right) using VBAKF in the simplest possible situation of a known static noise.

When the true covariance is known and static – an unlikely situation in the real world systems – we do not expect any improvements in predictions over CKF. Actually, its performance may be degraded trying to adjust to the noise, making the overall accuracy score worse than CKF. In the case when the covariance is known and changing, then it depends on how is one able to tune the algorithm parameters (such as ρ or τ), how big the changes are and how precise is the known value.

dát do conclusion, že VBAKF si ani tehdy nevedl moc špatně

Therefore, the main advantage is in the case when one do not know the true values or we do not know the initial estimate (or both), as is often the case in reality. It is the scenario where we expect VBAKF to surpass CKF, since if we are at least somewhat close to the initial estimate, it should *correct* to more precise values over time. Additionally, if the covariance is slowly changing, than VBAKF is able to adjust it as well.

4.3.1 Static noise

As is clear from the Figure 4.5 on page 31 , the claim mentioned in the previous section is validated. The CKF variant is slightly outperforming VBAKF, although both variants are performing quite well. Figure 4.6 on page 32 and Figure 4.7 on page 32 allows for some introspection to the observed behavior. VBAKF algorithm seems to pay the price for being able more flexible than CKF and it is following noise peaks too much. This could be to some degree controlled by the τ hyperparameter, but not much without degrading performance in some other means, such as being to rigid, or coalesce with CKF.

Let also demonstrate the situation when the true covariance is not known and set to some reasonable estimate close to the true covariance. The remarkable property of being able to adjust to the true covariance has been confirmed, as can be seen from RMSE comparison in Figure 4.8 on page 33 , the performance is significantly better, specially in the unobserved components of the state vector. This is even more interesting when considering that the standard deviation of prediction error in Figure 4.8 on page 33 is significantly lower for VBAKF than in CKF. This is the remarkable property of the proposed algorithm and figures for the first component in Figure 4.9 on page 33 and third component in Figure 4.10 on page 34 only supports that. Both algorithms started with noise covariance estimate which was off, but VBAKF variant could

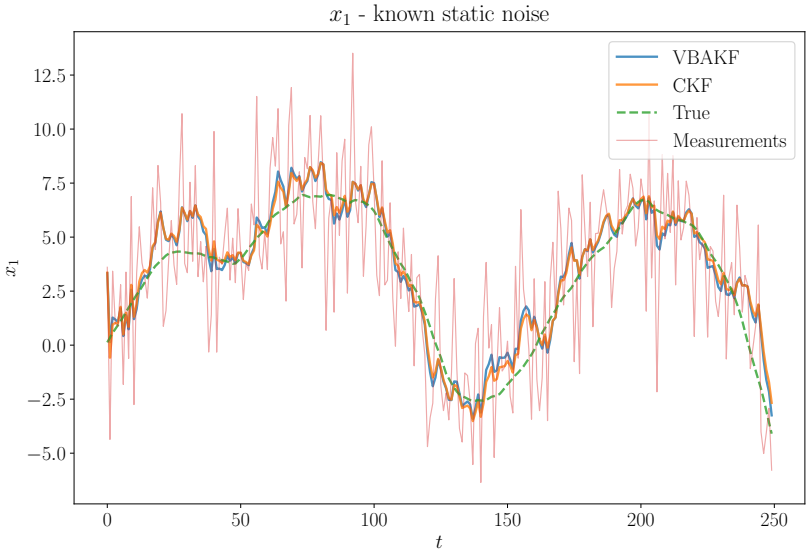


Figure 4.6: Performance of both filters for x_1 together with measurements and the true trajectory.

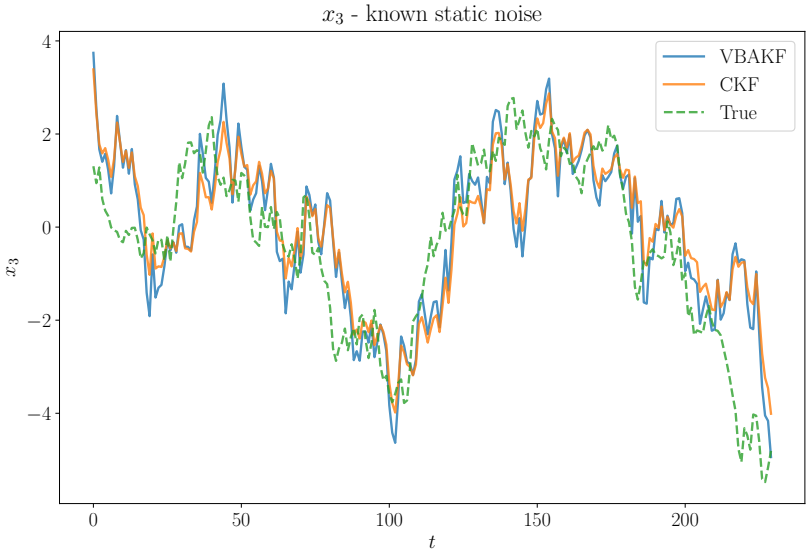


Figure 4.7: Performance of both filters for x_3 together with the true trajectory. This is purely estimated variable and not directly observed, hence the measurements are missing.

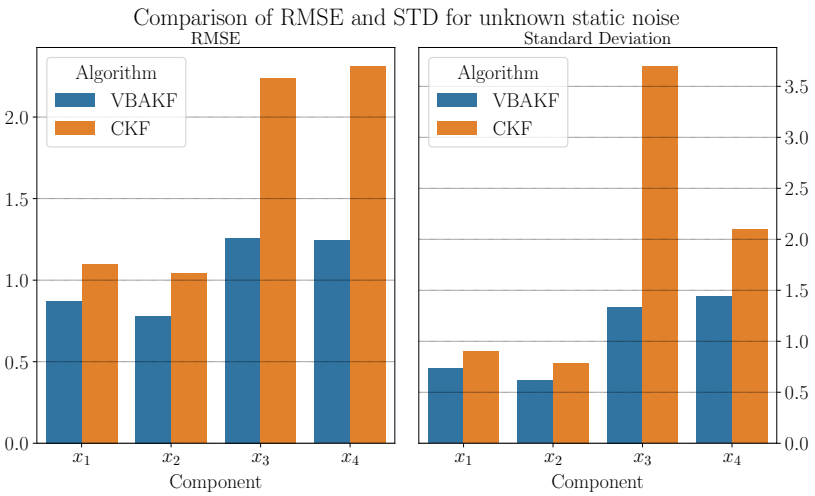


Figure 4.8: Although the performance gain in prediction errors of the observed components is already high (˜20%), the major difference is present in the unobserved ones, VBAKF giving over 2 times better results in terms of RMSE and 3 times better in terms of standard deviation.

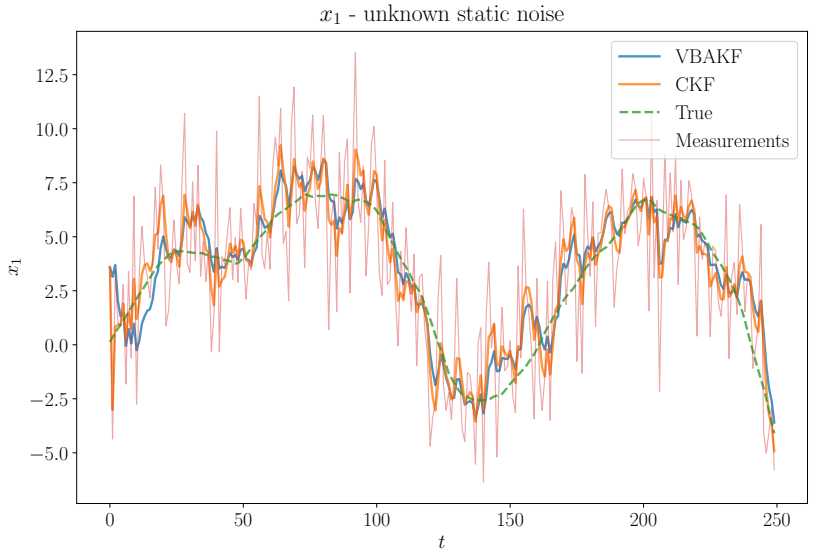


Figure 4.9: First and second component in the first row are observable, while the third and fourth is not.

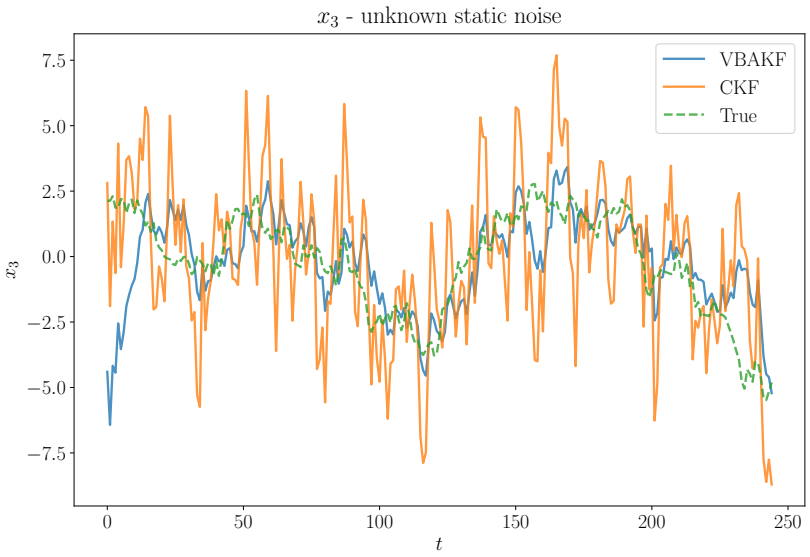


Figure 4.10: Estimated values of x_3 component by both algorithms and with true trajectory.

adjust for it, placing less emphasis on the measurements and believing more to the prediction step. CKF algorithm just stays on the same level of reliance with correction.

4.3.2 Variable noise

In this scenario, it is again considered that the value for true covariance is unknown and contrary to the previously discussed static case, it is slowly changing as is visualized on the Figure 4.4 on page 30. From the performance evaluation perspective, the situation is analogous to the previous case when the noise covariance is not known. Based on the expectations from the theory and the previous results, there is no surprise that VBAKF is performing significantly better over CKF with details in Figure 4.11 on page 35 . As can be seen from trajectories on Figure 4.12 on page 35 and Figure 4.13 on page 36 , VBAKF is again able to put less emphasis on the measurements during epochs of high noise, contrary to CKF being to sensitive in during these.

Finally, a very nice visualization of VBAKF capabilities can be seen on the graph Figure 4.14 on page 36. The adaptation of the noise covariance matrix to the true unknown covariance matrix is supporting the earlier findings of being able to account for noisy epochs.

4.3.3 Conclusion

The simulations shows that our single node VBAKF implementation is behaving as the theory predicts and in concordance with the reference source [30] where more emphasis on noise analysis were done. VBAKF does not perform better in case of a static noise when the covariance is known, merely just adding more computational time and possibly worse results trying to adjust to random fluctuations. It outperforms CKF in situations when the covariance of the noise is either unknown and, or, slowly changing. The performance of VBAKF was consistently better than CKF in these over many iterations and many variant of a noise. It is also worth mentioning that a very little tuning had to be done to get to these results. How much close the initial estimate must be to guarantee divergence is of course almost impossible to say in the real world cases, but in our testing case, even extreme initialization lead to quickly correcting the state

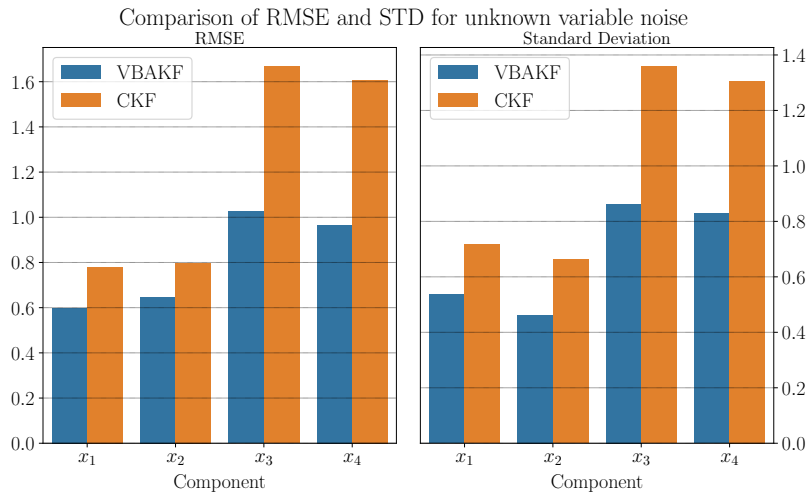


Figure 4.11: Similarly to the unknown static noise case, VBAKF is clearly performing significantly better in terms of RMSE (left) with much less variance in the estimation error (right). Furthermore, this is exhibited over all components.

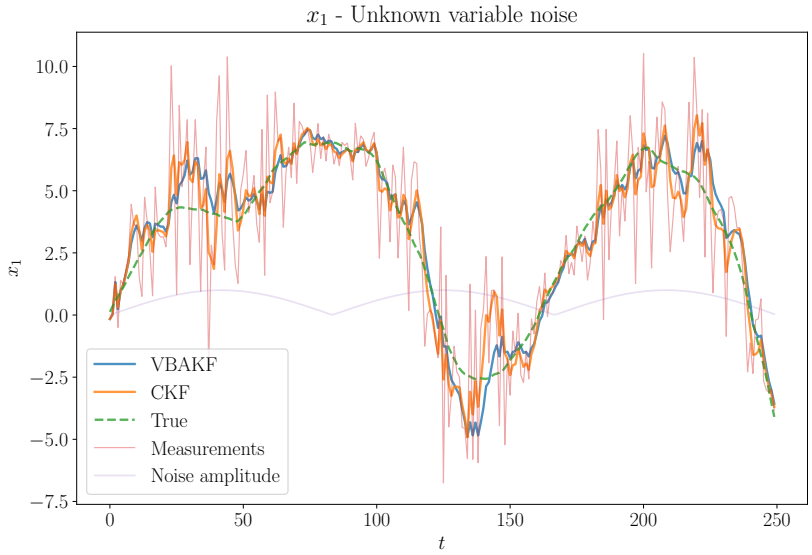


Figure 4.12: First component estimates for both algorithms with observations, true values and also the noise amplitude. Notice how VBAKF is becoming more rigid during high noise peaks while still being flexible to catch up to the trajectory change.

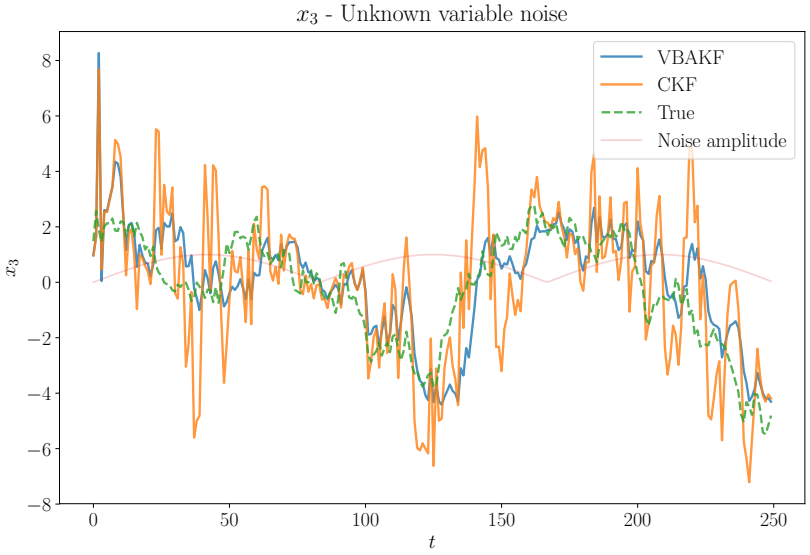


Figure 4.13: Third component estimates for both algorithms with observations, true values and also the noise amplitude. As in the case of Figure 4.12 on page 35, the VBAKF can control during very noisy intervals.

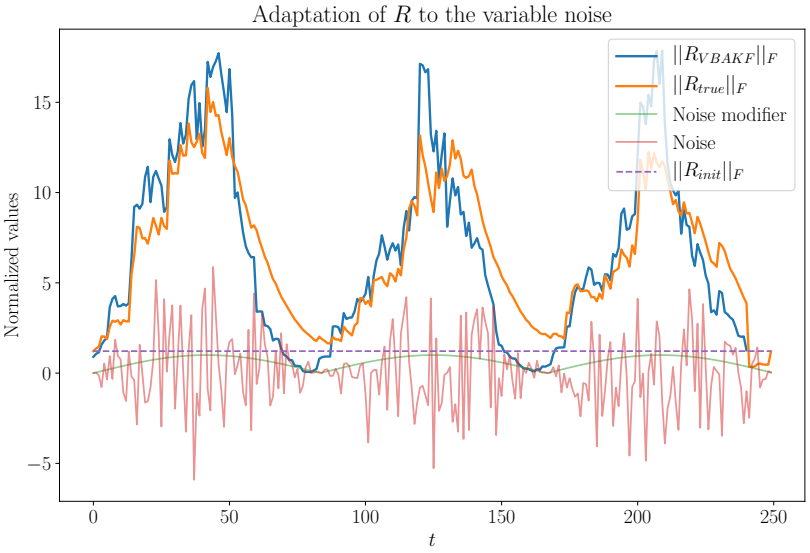


Figure 4.14: Frobenius norms comparison of the VBAKF algorithm and covariance of moving window (size 20 steps to the future) of the noise. The diagonal values of $\mathbf{R}$ are increasing during noise peaks, while going down during low noise epochs. The scale in this chart is not important, it is the trend of the norms. Those should be ideally as close as possible.

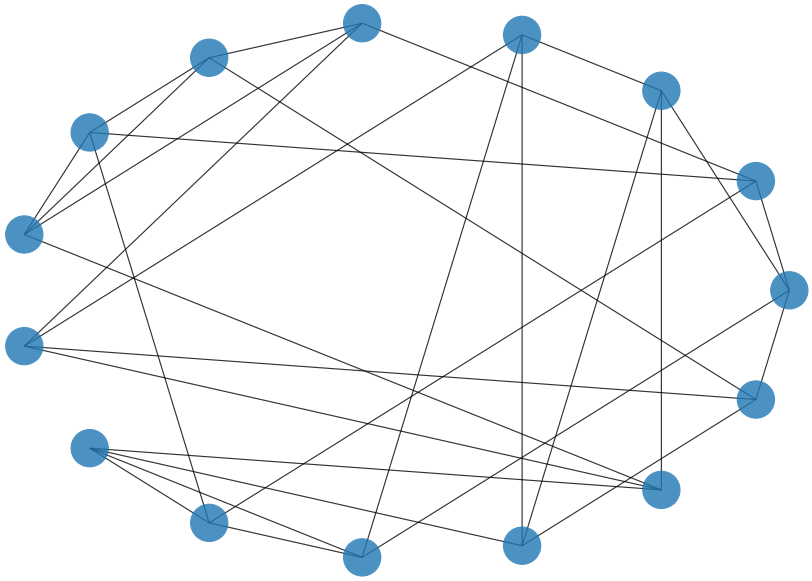


Figure 4.15: A network representing connection of different measurement nodes connected with the neighbors. The topology is a 4-regular graph.

(in less than 5 iterations). Such an extreme initialization were for example over three orders of magnitudes of the reality – a situation in which the CKF just does not provide any useful information.

4.4 Distributed VBAKF

Based on the results from the previous chapter 4.3, only a slowly changing noise is going to be considered to not clutter the analysis of VBAKF with diffusion versus VBAKF without diffusion. As has been mentioned on the beginning of this whole chapter, the figures when plotting information from multiple nodes should be read as a general trend exhibited by the system. It would be a mistake try to interpret them line by line.

4.4.1 Setting

Our network topology of nodes is for simplicity a 4-regular graph as is in Figure 4.15 on page 37 . A slowly changing, different noise is attributed to the each node in the network (Figure 4.16 on page 38). When we compare diffused versus non-diffused variant, the noise attributed in the both runs was the same in corresponding nodes to make the results comparable and reproducible. The results are averaged over at least 100 iterations.

4.4.2 Estimation performance

As expected, the estimation performance of the diffused variant is better than non-diffused variant. This is not only a result in the averaged results on the figure Figure 4.17 on page 38 , but it was consistent across literally all runs. To recapitulate – there was not a *single* run where would the non-diffused variant outperformed the diffused one in matter of Mean RMSE and standard deviation in Figure 4.17 on page 38. There is a price in time execution, since the diffusion variant does an additional step – diffusing parameters – in each iteration. In our

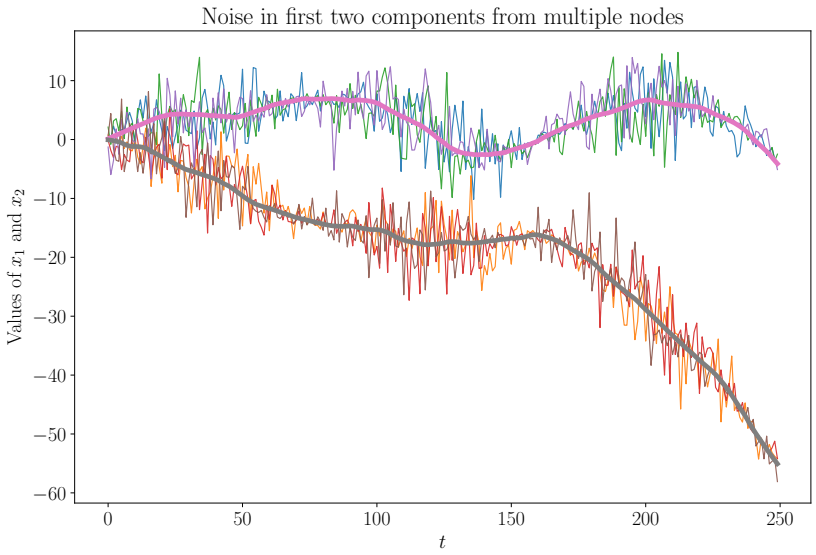


Figure 4.16: Values of first and second component with noise from different nodes.

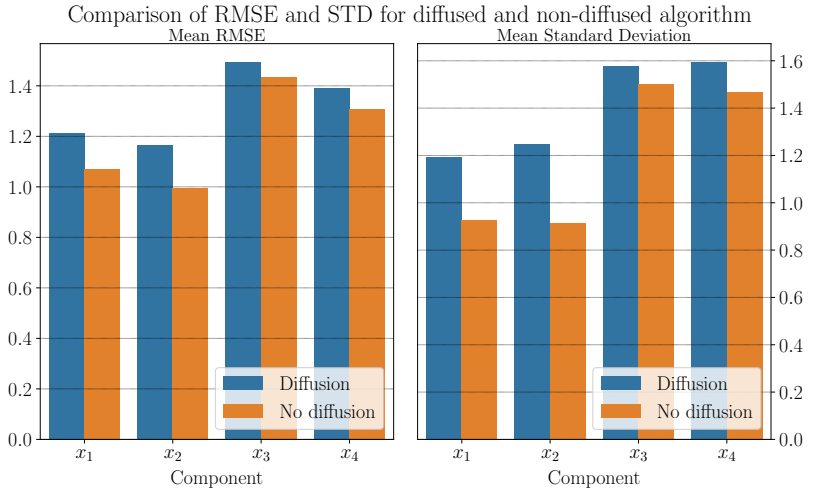


Figure 4.17: Diffusion variant outperforms the non-diffused variant not just in Mean RMSE (the left figure), but also in Mean Standard deviation over all runs (the right figure).

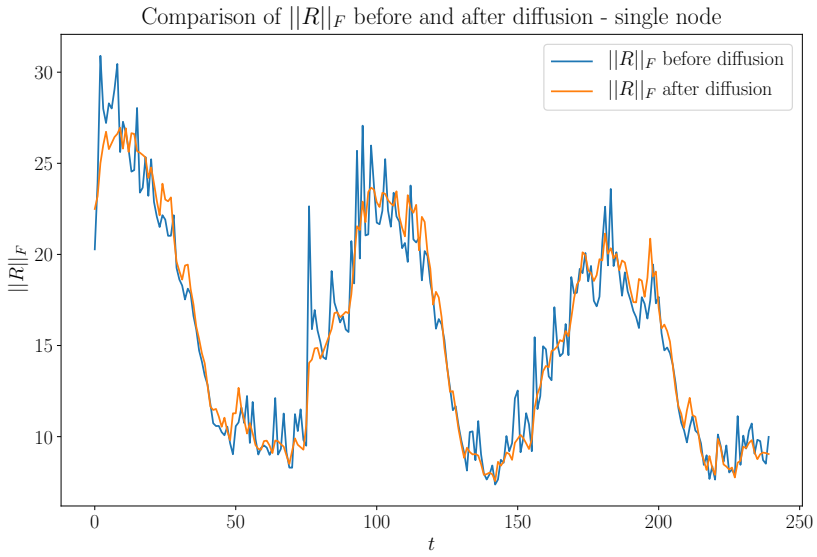


Figure 4.18: The diffusion algorithm smooths the trajectory for this single randomly chosen single node from the network, ignoring significant peaks which would otherwise led to inferior estimates.

setting, this was almost negligible, taking only about 5% longer than the non-diffused variant. This time penalty would of course depend on the chosen diffusion strategy.

4.4.3 Noise covariance analysis

The idea behind diffusion optimization is that the *averaging* over several nodes will eliminate extremes and smooths the estimation. This is clearly happening, as is clear from Figure 4.18 on page 39 , where we plot a Frobenius norm of the noise covariance matrix before and after diffusion step. Naturally, this phenomenon was exhibited on all nodes in the network, visualized in Figure 4.19 on page 40 .

Another view of this is presented in Figure 4.20 on page 40 and Figure 4.21 on page 41 . While there is almost no or very little pattern in the non-diffused variant, the diffusion variant got a general trend.

It is also interesting to note that in our simulations, the diffused variance exhibit much less divergence cases, which can be explained by the fact that any extremes are much likely to be suppressed by neighbor nodes participating in diffusion. An example can be seen on Figure 4.22 on page 41 .

4.4.4 Conclusion

Indeed, the diffusion extension to the VBAKF algorithm clearly performs better than variant without diffusion in place. This is exhibited consistently over many iterations with different initialization noise. It is relatively robust to the parameter setting when sane defaults are used (see [30] for more information).

The disadvantages of the diffusion algorithm are mostly practical limitations of real life applications.

First of all, a communication must be established between the communicating nodes. The implementation of this communication may not be possible, nodes can be for example only

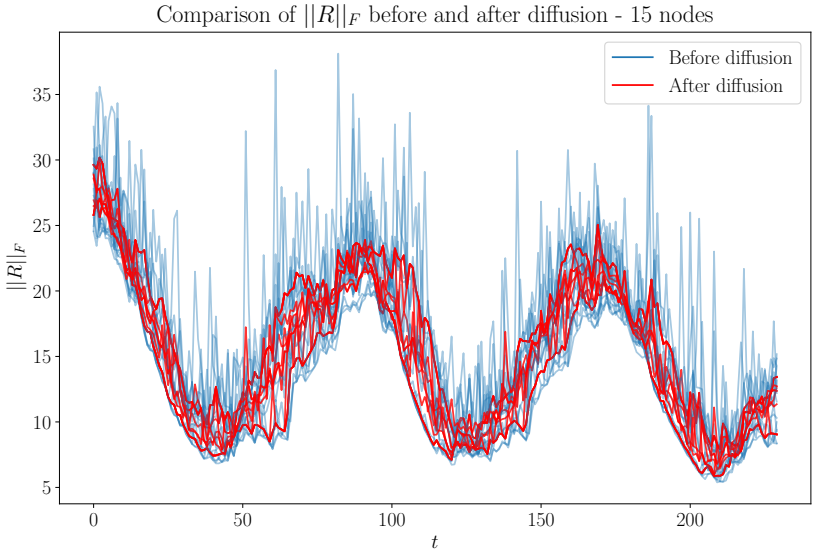


Figure 4.19: All nodes exhibit the behavior of smoothing out the noise covariance estimates, leading to the better results.

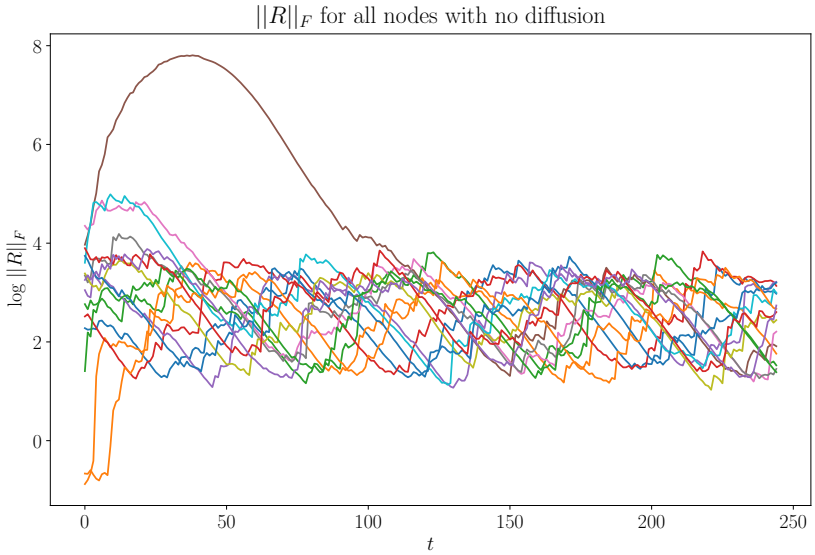


Figure 4.20: Nodes in non-diffusion variant do not exhibit any common pattern. There is also one of the nodes diverging quite significantly, considering the logarithmic scale. The diffusion variant do not exhibit such divergence.

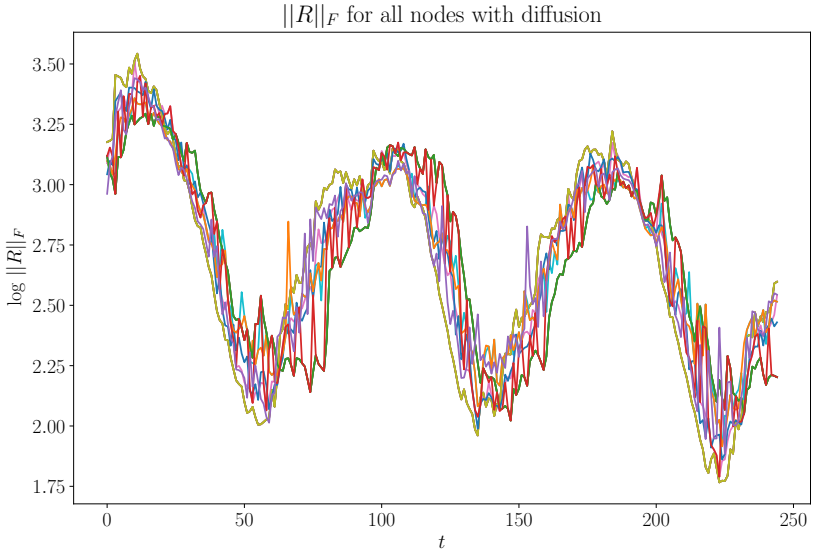


Figure 4.21: Nodes in the diffusion variant are able to pick up the general noise trend and "agree" on it to some degree, even though it may not be the best estimate for the underlying nodes.

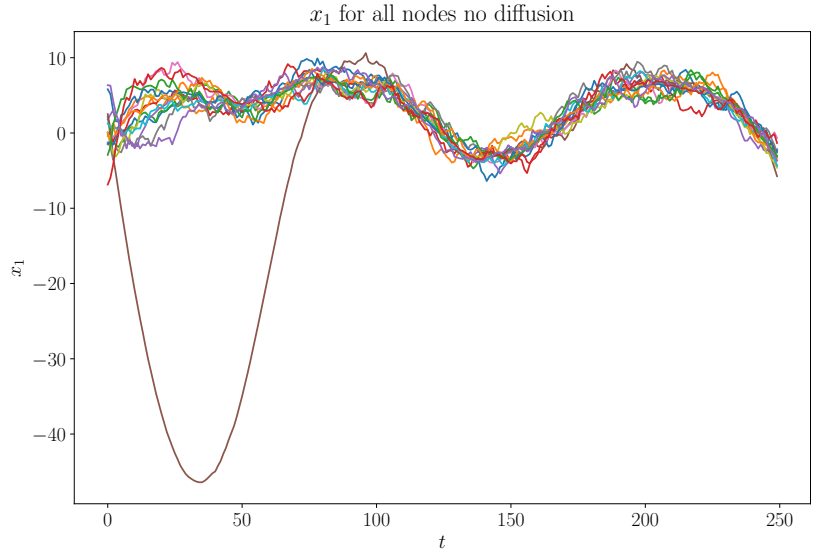


Figure 4.22: An example of diverging node present in the non-diffused variant.

connected to a master node or do not have ability to receive any information from the outer world, only to send one. Notwithstanding, this is not an intrinsic property of the method, but rather a common requirements for many diffusion and consensus approach. It is worth to mention that the algorithm does not rely on having all nodes participating in all time steps. As a matter of fact, the worst case scenario is a fallback to the default non-diffused variant when no nodes are communicating. This fact also facilitates the possible inclusion into an already existing solutions, since it can be implemented gradually, proceeding with the diffusion extension only when it is possible and a preliminary results shows effectiveness.

Secondly, there is additional implementation complexity, but as can be seen from the implementation, it is very easy to add the diffusion step into the existing algorithm. The diffusion strategy may be of course much more complex than it is in our case, but again, that is not an intrinsic property of the proposed algorithm but rather specific to the implementation.

In conclusion, when the node communication is not an issue, we believe that the proposed algorithm can be implemented for a very little price, while not performing worse than a traditional non-diffused variant.

4.5 Future work

It is obvious that this work does not exhaust the topic in its entirety. To the best of the author's knowledge, there has been a little done in the field of combining parameter diffusion or consensus algorithms with variational Bayes methods. The key questions regarding the algorithm behavior, such as the numerical stability or convergence criteria remains unanswered, unfortunately. Additionally, we believe that it would be a beneficial contribution to the field, if at least following were studied:

- What role network's topology plays and how to design networks which are cheap, but still effective.
- Sensitiveness to different kind of noise or its evolution.
- What are the most effective diffusion strategies in this setting.
- Sensitiveness to different state models.

Conclusion

The rise of cheap while powerful devices equipped with many sensors connected in a communication network justifies the necessity of developing efficient algorithms processing the network's information flow. It is no longer problem of not having the data, but rather generating useful insights based on them.

The algorithm studied in this work is an effective, cheap and elegant improvement of many real world scenarios where adaptive filtering is being used and noise covariances are unknown and changing. Using exponential family forms, it is relatively simple to understand and implement even into existing environments, enabling gradual, step-by-step adaptation. It does not suffer from any obvious drawbacks. Its only real disadvantage – having a communication protocol over nodes available – is a common requirements for many distributive algorithms and is not an intrinsic property of this one. It is also quite robust to the initial parameters and different kinds of noise, making it an excellent choice in many complex use cases. The nature of algorithm also enables good introspection and interpretability of the whole model, an important and often overlooked feature.

The simulation results strongly supports these claims and hopefully offers sufficient explanations for the observed behavior. The author hopes that even though this thesis delivers rather limited view of the method, it is still a valuable entry point for any further study on this matter.

Bibliography

[1] E. T. Jaynes. "Probability Theory: The Logic of Science." In: *The Mathematical Intelligencer* 27.2 (2003), pp. 83–83. ISSN: 0343-6993. DOI: 10.1007/BF02985800. arXiv: arXiv:1011.1669v3.

[2] Edward N ed Hájek Zalta. "Interpretations of Probability". In: *The Stanford Encyclopedia of Philosophy*. 2012. ISBN: 1095-5054. URL: https://plato.stanford.edu/.

[3] Maurice J Dupré and Frank J Tipler. "New Axioms for Rigorous Bayesian Probability". In: *Bayesian Analysis* 1.1 (2004), pp. 1–8. URL: http://dauns.math.tulane.edu/%7B~%7Ddupre/OLDPUBLIC/20090804%20BA%20FNL%203.pdf.

[4] Charles Friedman. "The Frequency Interpretation in Probability". In: *Advances in Applied Mathematics* 23.3 (1999), pp. 234–254. ISSN: 0196-8858. DOI: https://doi.org/10.1006/aama.1999.0653. URL: http://www.sciencedirect.com/science/article/pii/S019688589990653X.

[5] R T Cox. "Probability, Frequency and Reasonable Expectation". In: *American Journal of Physics* 14.1 (1946), pp. 1–13. DOI: 10.1119/1.1990764. URL: https://doi.org/10.1119/1.1990764.

[6] B de Finetti. *Theory of Probability: A critical introductory treatment*. J. Wiley & Sons, Inc., New York, 1975.

[7] E.T. Jaynes. "Bayesian Methods: General Background". In: *Maximum Entropy and Bayesian Methods in Applied Statistics* August 1984 (1986), pp. 1–25. DOI: 10.1017/CBO9780511569678. URL: /chapter.jsf?bid=CBO9780511569678A007%7B%5C&%7Dcid=CBO9780511569678A007.

[8] A Stuart and K Ord. "Kendall's Advanced Theory of Statistics: Volume I—Distribution Theory". In: 1994.

[9] Dirk P. Kroese and Joshua C.C. Chan. *Statistical modeling and computation*. 2014, pp. 1–400. ISBN: 9781461487753. DOI: 10.1007/978-1-4614-8775-3.

[10] Peter McCullagh. "What is a statistical model?" In: *The Annals of Statistics* 30.5 (2002), pp. 1225–1310. ISSN: 00905364. DOI: 10.1214/aos/1035844977.

[11] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Vol. 53. 9. 2013, pp. 1689–1699. ISBN: 978-0-387-31073-2. DOI: 10.1117/1.2819119. arXiv: arXiv:1011.1669v3.

[12] G. Upton and I. Cook. *A Dictionary of Statistics*. 2008, p. 512. ISBN: 978-0-19-954145-4. DOI: 10.1093/acref/9780199541454.001.0001.

[13] Christopher M Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN: 0387310738.

[14] Kamil Dedecius and Vladimira Seckarova. "Factorized Estimation of Partially Shared Parameters in Diffusion Networks". In: *IEEE Transactions on Signal Processing* 65.19 (Oct. 2017), pp. 5153–5163. ISSN: 1053-587X. DOI: 10.1109/TSP.2017.2725226. URL: http://ieeexplore.ieee.org/document/7973017/.

[15] David M Blei, Alp Kucukelbir, and Jon D Mcauliffe. "Variational Inference: A Review for Statisticians". In: (2017). arXiv: arXiv:1601.00670v6. URL: https://arxiv.org/pdf/1601.00670.pdf.

[16] Giorgio Parisi. *Statistical Field Theory*. 1988.

[17] Greg Welch and Gary Bishop. "An Introduction to the Kalman Filter". In: (2006). URL: https://www.cs.unc.edu/%7B~%7Dwelch/media/pdf/kalman%7B%5C_%7Dintro.pdf.

[18] Shih-Tin Lin. "Force Sensing Using Kalman Filtering Techniques for Robot Compliant Motion Control". In: *Journal of Intelligent and Robotic Systems* 18.1 (1997), pp. 1–16. ISSN: 09210296. DOI: 10.1023/A:1007946400645. URL: http://link.springer.com/10.1023/A:1007946400645.

[19] David Gaylor and E. Glenn Lightsey. "GPS/INS Kalman Filter Design for Spacecraft Operating in the Proximity of International Space Station". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Reston, Virigina: American Institute of Aeronautics and Astronautics, Aug. 2003. ISBN: 978-1-62410-090-1. DOI: 10.2514/6.2003-5445. URL: http://arc.aiaa.org/doi/10.2514/6.2003-5445.

[20] Gurnain Kaur Pasricha. "M P RA Kalman Filter and its Economic Applications Kalman Filter and its Economic Applications". In: (2006). URL: http://mpra.ub.uni-muenchen.de/22734/.

[21] Ingvar Strid and Karl Walentin. "Block Kalman Filtering for Large-Scale DSGE Models". In: *Computational Economics* 33.3 (Apr. 2009), pp. 277–304. ISSN: 0927-7099. DOI: 10.1007/s10614-008-9160-4. URL: http://link.springer.com/10.1007/s10614-008-9160-4.

[22] S. Gannot. "Speech processing utilizing the Kalman filter". In: *IEEE Instrumentation & Measurement Magazine* 15.3 (June 2012), pp. 10–14. ISSN: 1094-6969. DOI: 10.1109/MIM.2012.6204866. URL: http://ieeexplore.ieee.org/document/6204866/.

[23] Daniel M. Wolpert and Zoubin Ghahramani. "Computational principles of movement neuroscience". In: *Nature Neuroscience* 3.Supp (Nov. 2000), pp. 1212–1217. ISSN: 10976256. DOI: 10.1038/81497. URL: http://www.nature.com/doifinder/10.1038/81497.

[24] Magdi S Mahmoud and Haris M Khalid. "Bibliographic Review on Distributed Kalman Filtering". In: (2013). URL: http://cogprints.org/8906/1/MsM-KFUPM-DCC-2D-One%7B%5C%%7D5B3R%7B%5C%%7D5D.pdf.

[25] Felix Govaers. "Distributed Kalman Filter". In: *Kalman Filters - Theory for Advanced Applications*. InTech, Feb. 2018. DOI: 10.5772/intechopen.71941. URL: http://www.intechopen.com/books/kalman-filters-theory-for-advanced-applications/distributed-kalman-filter.

[26] Ian F. Akyildiz and Mehmet Can Vuran. *Wireless Sensor Networks*. 2010, p. 516. ISBN: 9780470515181. DOI: 10.1002/9780470515181. URL: http://scholar.google.com/scholar?hl=en%7B%5C&%7DbtnG=Search%7B%5C&%7Dq=intitle:No+Title%7B%5C#%7D0%7B%5C%%7D5Cnhttp://doi.wiley.com/10.1002/9780470515181.

[27] Mark E. Campbell and Nisar R. Ahmed. "Distributed Data Fusion: Neighbors, Rumors, and the Art of Collective Knowledge". In: *IEEE Control Systems* 36.4 (Aug. 2016), pp. 83–109. ISSN: 1066-033X. DOI: 10.1109/MCS.2016.2558444. URL: http://ieeexplore.ieee.org/document/7515322/.

[28] Ali H Sayed. "Diffusion adaptation over networks". In: (2013). arXiv: arXiv:1205.4220v2. URL: https://arxiv.org/pdf/1205.4220.pdf.

[29] Reza Olfati-Saber. "Distributed Kalman Filtering and Sensor Fusion in Sensor Networks". In: (). URL: https://pdfs.semanticscholar.org/fd77/043a3e588e03028589615b51f51060074a5b.pdf.

[30] Yulong Huang et al. "A Novel Adaptive Kalman Filter With Inaccurate Process and Measurement Noise Covariance Matrices". In: *IEEE Transactions on Automatic Control* 63.2 (Feb. 2018), pp. 594–601. ISSN: 0018-9286. DOI: 10.1109/TAC.2017.2730480. URL: http://ieeexplore.ieee.org/document/8025799/.

[31] Maurice G. (Maurice George) Kendall et al. *Kendall's advanced theory of statistics.* Edward Arnold, 2004. ISBN: 0340807520. URL: https://eprints.soton.ac.uk/46376/.

[32] Simo Särkkä Jouni Hartikainen. "Variational Bayesian Adaptation of Noise Covariances in Non-Linear Kalman Filtering". In: (Feb. 2013). arXiv: 1302.0681. URL: http://arxiv.org/abs/1302.0681.