Correlated Estimation Problems and the Ensemble Kalman Filter

Jan Čurn

A Dissertation submitted to the University of Dublin, Trinity College

in fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

October 2014

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Jan Čurn

Dated: October 12, 2014

Acknowledgements

I cannot believe I am writing this. I have been waiting for this moment so long, and now when it's here, I don't even know where to start.

It is exactly five years since I landed in this wonderful country of Ireland and began the work, which ensued in this PhD thesis. It was a leap into the dark. I knew nothing about research, had never left my hometown for more than a few weeks, didn't know a single person here and couldn't even speak English very good. Now I know this was the best decision I have ever taken. Next week I'm moving away, so I will use this very special one-off opportunity to thank the people who made this journey such an incredible, truly life-changing, experience. Because of you, yet again it feels like I'm leaving home.

Foremost, I have to thank my supervisor, Vinny Cahill, for introducing me to research, his academic guidance during all those years, and also, for securing generous funding for most of that time. Vinny, your help with this thesis was invaluable.

A special thank you goes to Dan Marinescu, a fellow PhD candidate, my research colleague, co-author of papers, flatmate, and a travelling buddy. Dan, thanks to you now I have an idea what a marriage must look like. If things go well, we will graduate, together.

I am extremely grateful to all the current and former members of the Distributed Systems Group who made this College and Dublin in general such a great place to be. In particular, thank you Tim Walsh for being always available for a pint or two and listening to all my PhD-related rants. Neil "Coco" O'Connor and Jenny Munnelly for being such a great compact unit of friends and for adopting Shiu Lun Tsang as your own. Andronikos Nedos for evoking in me a deep admiration for feta cheese and tzatziki. Raymond Cunningham for giving me a whole new perspective on post-game football analyses. Mithileash Mohan and Guoxian Yang for their very precious sense of humour. Marco Slot for always being instantly ready to go for a coffee (Marco, we drank about a thousand of them). Sean Blanchfield for introducing me to Dublin's vibrant technology startup scene. As'ad Salkham for proving to me that male pattern baldness

can be dealt with with pride. Marcin Karpiński for providing an early warning on what this does to people (it was worth it!). Last Simone Grassi, you are the true inspiration. We all miss you. Also, I would like to thank to all the people in the College and elsewhere for reminding me every day what I was actually doing by mentioning those three magic letters in every context possible - P H D.

Of course, I must thank to the whole "lobos" group, you guys really showed me there is an alternative way of travelling to places and meeting people. Most of us now live in different cities and others are to follow (except of Julian Waldthaler, the future Lord Mayor of Dublin), let's make an advantage out of it. The world is much smaller than it used to be. Specifically, thank you Hassan Yassine, Henri Nadir Nekmouche, and Georgios Fanis, for being the coolest flatmates I could have ever imagined. I could write a book about all the events that took place in P128, although I'm not sure who'd be the audience. Also thank you for all the free lunches at Google.

I also want to thank to all my friends in Prague who always supported me and welcomed me back as if I never left. Sorry, you are too many to name.

The final part of these acknowledgments is dedicated to my family, whose infinite support, encouragement and love was essential during all these and other years. Mum, thank you for everything. Thank you my sister, Zuzana Horáčková and brother, Jakub Horváth, for confirming the law of nature that every family is somewhat weird. Thank you my grandfathers, František Čurn and Vladimír Horáček, for sparking technical curiosity in the little boy. Thank you my grandmothers, Růžena Čurnová, for teaching me responsibility, and Marcela Horáčková, for teaching me manners. Lastly, thank you my father, Petr Čurn, you gave me more than you'd have though, and keep affecting my decisions long after you're gone.

I thank you all, you made me who I am.

Jan Čurn

University of Dublin, Trinity College

June 30, 2013

Abstract

The Kalman filter is a recursive algorithm that estimates the state of a linear dynamical system from a sequence of noisy sensor measurements. Due to its relative simplicity, numerical efficiency and optimality, the Kalman filter and its variants have been applied to a wide range of problems in technology, notably in the areas of guidance, navigation, and control. The traditional definition of the Kalman filter is based on the assumption that at any given time, the errors associated with the predicted state estimate and the observation are statistically independent. However, in many practical problems, this assumption is not satisfied, and as such the Kalman filter may provide overconfident state estimates and diverge. This can have serious consequences in the context of safety-critical systems.

Although there are modifications of the Kalman filter that accommodate various types of correlation in the process and observation noises, these are not suitable in the situation where the correlation between the errors associated with the predicted state estimate and the observation is caused by the presence of common past information between the state estimate and the observation, which is characteristic of distributed sensor networks. On the contrary, existing methods that deal with the common past information problem either provide overly conservative estimates, or have too strict assumptions on the structure of the problem, such as the communication topology of the sensor network.

This thesis presents two new filters to address various correlated estimation problems that are based on the Ensemble Kalman filter, a Monte Carlo variant of the Kalman filter, which represents the state estimates and observations using sets of random samples instead of the conventional mean vectors and covariance matrices. Specifically, both of these filters provide a new generalised update rule that computes consistent state estimates even in the presence of correlation between the errors associated with the state estimate and the observation. This is only possible due to the fact that in the context of the Ensemble Kalman filter, the magnitude of such a correlation can be estimated from the random samples.

The new filters retain all of the important features of the Ensemble Kalman filter, such as scaling linearly with the number of state-space dimensions, and supporting non-linear process and observation models. An analysis of the numerical properties of the filters is provided, including a comparison with state-of-the-art methods in several benchmark scenarios. Furthermore, in order to demonstrate their practical utility, the new filters have been applied to three different real-world problems in the larger field of robot localisation: cooperative vehicle localisation, simultaneous localisation and mapping, and global satellite-based positioning.

Publications Related to this Ph.D.

Čurn, J., Marinescu, D., O'Hara, N., and Cahill, V. (2013). Data incest in cooperative localisation with the Common Past-Invariant Ensemble Kalman filter. In *Proceedings of the 16th International Conference on Information Fusion (FUSION 2013)*. To appear.

Čurn, J., Marinescu, D., Lacey, G., and Cahill, V. (2012b). Estimation with non-white Gaussian observation noise using a generalised Ensemble Kalman filter. In *Proceedings of the IEEE International Symposium on Robotic and Sensors Environments (ROSE 2012)*.

Čurn, J., Marinescu, D., and Cahill, V. (2012a). On the Monte Carlo representation of uncertain spatial constraints. *Procedia Engineering*, 41(0):37–46. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).

O'Hara, N., Slot, M., Marinescu, D., Čurn, J., Yang, D., Asplund, M., Bouroche, M., Clarke, S., and Cahill, V. (2012). MDDSVsim: An integrated traffic simulation platform for autonomous vehicle research. In *The International Workshop on Vehicular Traffic Management for Smart Cities (VTM 2012)*.

Marinescu, D., Čurn, J., Bouroche, M., and Cahill, V. (2012). On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach. In *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems (ITSC 2012)*.

Čurn, J., Marinescu, D., and Cahill, V. (2010). A flexible approach to management and processing of collaborative vehicular perception data. In *Proceedings of the Workshop on Emergent Cooperative Technologies in Intelligent Transportation Systems (ECTITS 2010) at the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010)*.

Marinescu, D., Čurn, J., Slot, M., Bouroche, M., and Cahill, V. (2010). An active approach to guaranteed arrival times based on traffic shaping. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010)*.

Contents

List of Tables

List of Figures

List of Definitions

Chapter 1

Introduction

Observations of any physical phenomenon are inherently uncertain. Whenever a physical quantity is measured by a sensor, the observed value will be affected by two types of error: *systematic* error and *random* error. The systematic error describes the situation where the mean of a series of observations differs significantly from the observed quantity, and typically can be eliminated by suitable sensor calibration. The random error, on the other hand, is caused by effects that cannot easily be predicted and eliminated, such as sensor vibrations, electric fluctuations in the measurement instrument, atmospheric effects, environmental radiation, or at a very low level, the quantum nature of our universe with Copenhagen interpretation (Wimmel, 1992), hoping the universe is not superdeterministic.

Even though the random error, also known as the *noise*, cannot be fully eliminated from the measurements, it is possible to reason about the physical quantity being observed in a probabilistic fashion. Assuming a certain knowledge of the sensor properties, the measured value effectively provides an estimate of the true value of the physical quantity, with an associated probability distribution. A fundamental insight is that by combining a number of noisy measurements from one or more sensors, the uncertainty of the estimate can be significantly reduced.

Over the past half century, sophisticated probabilistic algorithms have been developed to estimate the true values of physical quantities, the so-called *system state*, from a sequence of noisy sensor measurements. These algorithms vary in their assumptions on the properties of the measurement noise, its probability distribution, and the model of the underlying physical system dynamics. Most notably, the Kalman filter, originally proposed in Rudolf Emil Kálmán's landmark paper (Kalman, 1960), deals with a particular case where the physical system can be

modelled as a discrete-time linear dynamical system perturbed by a Gaussian noise, the observations represent linear combinations of the system state variables, and the physical measurements are affected by a Gaussian noise. It happens that these assumptions are not overly restrictive, and many real-world physical systems can be modelled and estimated using the Kalman filter, or one of its variants, with sufficient accuracy.

From the computational perspective, the Kalman filter is a recursive algorithm that repeats two steps: the *prediction* and the *update*. The prediction step propagates the current state estimate to the next discrete time step using a *process model*, while potentially increasing the uncertainty of the estimate to compensate for the effects of the *process noise*. The update step assimilates a measurement, affected by the *observation noise*, in order to refine the state estimate, potentially reducing its uncertainty. The relation between the state estimate and the measurement is given by the *observation model*, a function that computes an expected measurement given the current state estimate. In fact, in the Kalman filter the updated state estimate is computed simply as a weighted average of the observation and the previous state estimate, with weights corresponding to the covariance of their errors. This weight factor is commonly referred to as the *Kalman gain*. The simplicity and efficiency of the algorithm, and the fact that it converges even if the system parameters are not precise, were the key factors why the Kalman filter has gained wide popularity and has been applied in all areas of technology, ranging from navigation of spacecraft to analysis of financial time series (Simon, 2006).

The traditional definition of the Kalman filter assumes that both the process noise and the observation noise are white and independent, implying that, at any time, the errors associated with the predicted state estimate and the observation are statistically independent, and therefore, uncorrelated. However, various types of correlation in the noises occur in many practical applications. For example, in radar tracking, an effect known as *target scintillation* causes the measurement error to have a finite bandwidth, and hence appear as a time-correlated sequence (Skolnik, 2008). Consequently, the error in the current state estimate then becomes correlated with the current measurement error. If such correlation is ignored, the filter may provide overconfident state estimates and diverge, which can lead to serious consequences in the context of safety-critical systems.

This thesis deals with the general problem of Kalman filtering in the presence of a correlation between the errors associated with the state estimate and the observation. Such a correlation can be caused by various factors, for example, by a time-correlated observation noise as described above, or simply by a correlation between the process noise and the observation noise.

Furthermore, the solution to the problem depends on the interpretation of the correlation. In particular, the situation, where the correlation is caused by the presence of *common past information* shared between the state estimate and the observation, requires a different treatment than the situation where it is caused by correlation in the noises. This common past information problem is often encountered in distributed sensor networks, where the nodes estimate the state of some local or global physical system, while communicating the estimates with each other. For example, consider node A that sent its state estimate to node B, which considered this estimate as an observation and assimilated it into its own local state estimate. Now, if node B sends its estimate back to node A, this estimate can no longer be considered as an independent observation, as it depends on A's local state estimate - their errors are correlated. In the literature, this problem is also referred to, less poetically, as *data incest*.

A number of approaches have been proposed to address the problem of linear state estimation with common past information. In the context of distributed sensor networks, methods such as the Decentralised Kalman filter (Durrant-Whyte et al., 1990), Channel filters (Grime et al., 1992) and Kalman Consensus algorithms (Olfati-Saber, 2009) address specific instances of the problem given various assumptions on the network topology and communication patterns. However, these methods have either overly restrictive assumptions on the network topology or they provide no guarantees on the consistency of their estimates. Moreover, these methods are specific to sensor networks, and they offer no generalisation of the Kalman filter update rule that would estimate the state in the presence of common past information. One such method, the Covariance Intersection algorithm (Julier and Uhlmann, 1997a), assumes a correlation between the errors associated with the state estimate and the observation is present but unknown, and hence provides only a conservative suboptimal update rule. On the other hand, the existing extensions of the Kalman filter that enable it to operate with various types of correlation in the process and observation noises, such as the Generalised Kalman filter, state augmentation, or measurement differencing (Simon, 2006), are not suitable to the problem of common past information.

1.1 Background

Although there is no formal definition, it is commonly understood that the term *Monte Carlo method* refers to any algorithm whose results depend on repeated random sampling. The origin of the Monte Carlo method dates back to 1946, when Stanisław Ulam considered what is the probability of winning a game of Canfield solitaire (Eckhardt, 1987). While deterministic

computation of this probability turned out to be difficult, Ulam speculated that if the game is played many times with randomly shuffled cards, one could estimate the probability simply by counting outcomes of particular games. This idea only became practical thanks to the recent advent of automatic computing machines - computers. Subsequently, Ulam developed the idea together with John von Neumann, and applied it to the analysis of neutron chain reactions in fission devices as part of the Manhattan project. A secret project required a code name, so von Neumann chose one after the Monte Carlo Casino in Monaco. Since that time, the Monte Carlo method has become an invaluable tool to many practical problems in all areas of science.

Due to its relative computational expense, for several decades since its discovery, the Monte Carlo method remained in the domain of off-line computations performed on contemporary super-computers. Nowadays, the situation is different as computers with sufficient performance are widely available, and Monte Carlo methods are routinely being applied in real-time systems, such as robots. For example, the *particle filter* (Doucet and Johansen, 2009), a popular non-linear non-Gaussian estimator that represents a state estimate using a set of random samples, is used for global localisation by real-world self-driving cars (Thrun and Urmson, 2011).

The Monte Carlo method also found its use in the context of linear state estimation, because it was found to resolve two important limitations of the Kalman filter. Firstly, the Kalman filter assumes that both the underlying system dynamics and the observation model are linear. Although the *Extended Kalman filter* (EKF) and especially the *Unscented Kalman filter* (UKF) (Julier and Uhlmann, 1997b), can alleviate the problem in many practical applications, both of these methods have a theoretical limit that prevents them from supporting any general non-linear models with arbitrary accuracy. Secondly, the computational complexity of the Kalman filter, as well as the EKF and UKF, scales no better than quadratically with the number of state-space dimensions (Wan and Van Der Merwe, 2000), which makes these filters inapplicable to problems with a large number of state-space dimensions, such as weather forecasting.

The limitations of the Kalman filter resulted in the development of the *Ensemble Kalman filter* (EnKF) - a Monte Carlo variant of the Kalman filter originally proposed in (Evensen, 1994). In the EnKF, the current state estimate is represented by a set of random Monte Carlo samples, organised in a structure known as the *ensemble*. Given a sufficient number of samples (ensemble members), the probability distribution of the estimate can be represented with an arbitrary accuracy, and it can be propagated through an arbitrary dynamic model. Similarly, observations are also represented using a set of random samples, thus enabling an accurate representation of potentially non-Gaussian probability distributions of the measurement noise.

Although, from a theoretical perspective, the basic Kalman filter assumption on the Gaussian nature of all the involved probability distributions has not been lifted, the EnKF is routinely being applied to highly non-linear non-Gaussian estimation problems, while providing estimates with potentially a higher accuracy than both the EKF and UKF. Additionally, both the space and time complexity of the EnKF scale linearly with the number of state-space dimensions. Due to these properties, the EnKF became a de facto standard tool for data assimilation in the Earth sciences, where the modelled systems are often non-linear and high dimensional (Evensen, 2009).

1.2 Motivation

The problem of linear state estimation in the presence of a correlation between the errors associated with the predicted state estimate and the observation is an important one, as it appears in many practical problems in technology. The known approaches to the problem are based around two principal interpretations of the origin of the correlation: correlated noises and common past information. Unfortunately, the existing extensions of the Kalman filter that address correlated noises are not applicable to the common past information problem, while methods addressing the common past information problem are either not general enough, or provide overly pessimistic or inconsistent estate estimates.

In particular, there is no generalisation of the Kalman filter that provides optimal state estimates in the presence of common past information between the state estimate and the observation. Clearly, in order to provide such estimates, the exact magnitude of the correlation between the associated errors needs to be known to the filter, which is, however, difficult to compute analytically in all but trivial systems. Therefore, no such generalisation of the Kalman filter has ever been developed, and only methods making special provisions to keep the track of the correlation in specific scenarios, or conservative methods ignoring the magnitude of the correlation altogether, are available. Treating the problem of common past information in the context of a single local filter would have important advantages - the algorithm is likely to be computationally efficient and simple to implement. These two properties proved over and over to be an important factor behind a successful adoption of many estimation algorithms, including the Kalman filter itself.

While attracting very little attention outside of the Earth sciences community, the EnKF possesses several unique features. From the perspective of this thesis, the most important feature

is the representation of state estimates and observations using ordered sets of random samples, instead of the conventional mean vectors and covariance matrices. With such a representation, it is possible to estimate the cross-covariance between errors associated with a state estimate and an observation, with an accuracy determined only by the number of samples. This opens the possibility to create a generalisation of the EnKF, which would operate in the presence of a correlation between the errors associated with the state estimate and observation. Therefore, this thesis will derive two new generalisations of the EnKF: the *Common Past-Invariant Ensemble Kalman filter* (CPI-EnKF) and the *Augmented EnKF*. Both of these filters provide consistent state estimates in the presence of correlation, regardless whether is caused by correlated noises or common past information. These two generalisations of the EnKF represent the main contribution of the thesis.

Both the CPI-EnKF and the Augmented EnKF filters are general methods that can be applied to a range of problems, even problems that might not appear to fall into the categories of estimation with common past information. Both filters retain all the important features of the EnKF, such as support for non-linear process and observation models, and runtime complexity linear in the number of state-space dimensions. Moreover, they both can also be applied to estimation problems with correlated noises; although the state estimates are suboptimal in this case, the computational benefits, stemming from the Monte Carlo nature of the filters, might advocate their application in certain scenarios. This thesis demonstrates the application of the new filters to three very distinct problems in the larger field of robotic localisation: cooperative vehicle localisation, simultaneous localisation and mapping and global satellite-based positioning. This work also challenges the status quo and demonstrates that the EnKF is a practical approach applicable to real-time estimation problems, in robotics and elsewhere.

1.3 Approach

In order to illustrate the correlated estimation problems outlined by this thesis, demonstrate the limitations of existing methods and thus motivate the development of the new EnKF-based filters, the thesis provides a detailed review of the state of the art in all related areas. The first part of the review, presented in Chapter 2, provides a formal definition of the Kalman filter, as well as all of its important variants: the Extended Kalman filter (EKF), the Information Filter, the Unscented Kalman filter (UKF) and the Ensemble Kalman filter (EnKF). Additionally, the chapter reviews the particle filter, a popular non-linear non-Gaussian estimator which is also

relevant to this thesis. The main focus lies on practical computational properties of the filters, such as their asymptotic space and time complexity, and support of non-linear models. Also, this chapter introduces a formalism which will be used in the rest of the thesis.

The second part of the state of the art review, presented in Chapter 3, deals with existing extensions of the Kalman filter that enable it to provide optimal state estimates even in the presence of various well-modelled types of correlations in the noises, specifically: a correlation between the process and observation noises, a sequentially correlated process noise, and a sequentially correlated observation noise. The case of a correlation between the process and observation noises is addressed by a method known as the *Generalised Kalman filter*, which defines a generalised Kalman filter update rule to account for the correlation. One the other, the cases of sequential correlation in the process or observation noise can be treated by an algebraic reordering of the conventional Kalman filter equations, leading to the approaches known as *state augmentation* and, in the case of sequentially correlated observation noise, *measurement differencing*. This chapter offers the most comprehensive review of all these methods in one place, as their description is somewhat scattered in the literature.

The third and final part of the state of the art review, presented in Chapter 4, deals with linear estimation problems affected by common past information, which are characteristic of distributed sensor networks. The review begins with a description of the most naive applicable methods - the *centralised Kalman filter* and the *Decentralised Kalman filter*, and follows with a detailed analysis of the origin of the common past information problem. Finally, the more involved methods addressing this problem are discussed, such as *Channel filters*, *Kalman Consensus algorithms*, and the *Covariance Intersection*, with a focus on their limitations that are to be addressed by the new generalisations of the EnKF developed in this thesis. Similarly to the case of correlated noises, the goal of the chapter is to provide the most comprehensive review of this area to date.

The main contribution of this thesis - the generalisations of the EnKF to address correlated estimation problems - is developed in Chapter 5. This is done in two steps. First, an extension to the prediction step of the EnKF is provided, which can accommodate for any type of correlation in the process noise, while avoiding the increased computational cost associated with state augmentation. Second, three alternative generalisations of the EnKF update rule are proposed: the *Generalised EnKF*, which is based on the Generalised Kalman filter, the *Common Past-Invariant Ensemble Kalman filter* (CPI-EnKF), which assumes that the correlation is caused exclusively by the presence of a shared error term, and the *Augmented Ensemble Kalman filter*

(Augmented EnKF), which is derived using a variant of the state augmentation procedure. All the update rules share a common goal to provide state estimates in the presence of correlation between the errors associated with the predicted state estimate and the observation, but each of the rules is based on a different presumptions.

Chapter 6 provides a comprehensive numerical evaluation of all the new methods developed in the previous chapter using canonical instances of all the correlated estimation problems presented in the state of the art review. The goal of the evaluation is to assess the applicability of the new methods to these problems, and to measure the accuracy and consistency of their estimates, also in comparison with the state-of-the-art methods. One of the results of the evaluation is that the Generalised EnKF provides different estimates than the Generalised Kalman filter, which effectively serves as a counterexample to the Generalised EnKF. On the other hand, the evaluation shows that both the CPI-EnKF and the Augmented EnKF provide reasonable and consistent state estimates in most of the estimation problems affected by correlated noises, while in the case of problems affected by common past information they even provide more accurate estimates than the only comparable state-of-the-art method - the Covariance Intersection algorithm. Additionally, this chapter provides an analysis of the accuracy of all the EnKF variants depending on the number of Monte Carlo samples, which is an important consideration for practical applications.

The new generalisations of the EnKF for correlated estimation problems, in particular the CPI-EnKF and the Augmented EnKF filters, were also evaluated in three distinct real-world applications in the larger field of robotic localisation. The goal of this evaluation, which is presented in Chapter 7, is to demonstrate a practical utility of the methods developed in this thesis.

The first practical problem considered is cooperative vehicle localisation, in which a group of vehicles is driving in an outdoor environment, while each of them is estimating its position using a global positioning system (GPS) and odometry. Additionally, the vehicles can improve their estimates by observing positions of other vehicles using a proximity sensor, such as a radar, and mutual communication, which is especially helpful to those vehicles operating in areas with no GPS coverage. In such a distributed fusion system, each vehicle needs to account for the fact that information received from other vehicles might originate in part from the vehicle itself, resulting in a correlation between the state estimate and observation errors - the common past information problem.

The second problem addressed in the evaluation is the well-known problem of simultaneous

localisation and mapping (SLAM), in which a mobile robot navigates in a previously unknown environment, while simultaneously building a map of that environment using its sensors. Although, after 20 years of intensive research of SLAM, a number of powerful algorithms have been proposed and, at least from a theoretical perspective, SLAM is considered as a solved problem (Durrant-Whyte and Bailey, 2006), there are some specific weaknesses of the state-of-the-art methods, such as computational inefficiency for life-long missions. The CPI-EnKF can be used in order to maintain an efficient representation of uncertain positions of spatial features in a robot's environment. Using the proposed approach, the spatial features can be queried and removed in constant time, and the representation can be updated in time linear in the number of spatial features. Therefore, such a representation can serve in life-long missions.

The third evaluation scenario considers a mobile robot that navigates through an outdoor environment and estimates its global position using observations from visual odometry and a stand-alone GPS receiver, whose readings are affected by a sequentially correlated noise process. This section proposes a novel model of the GPS error, which considers both the sequential correlation in the measurements error, as well as the horizontal dilution of precision reported by the GPS receiver. Although such an error model could not be supported using conventional approaches because of an unacceptable performance overhead, it can be supported in an estimation system built using the CPI-EnKF.

1.4 Contribution

This thesis contributes to the state of the art in the areas of state estimation and robotics with the following results:

- The new CPI-EnKF and Augmented EnKF filters derived in this thesis provide the only universal update rules applicable in the presence of a correlation in errors associated with the state estimate and the observation that provide better estimates than the Covariance Intersection algorithm.

- This thesis presents the only estimation system that leverages the unique Monte Carlo representation of the EnKF to address the problem of correlation.

- The state of the art review provides the most comprehensive analysis to date of linear state estimation methods operating in the presence of correlated noises, as well as correlation caused by common past information.

- The thesis demonstrates using a counterexample that the update rule of the Generalised Kalman filter is not applicable in the context of the EnKF, and thus the thesis establishes that none of the variants of the Kalman filter applicable to the problem of correlated noises is also applicable to the problem of common past information.

- The cooperative localisation system developed using the new EnKF-based estimation framework represents one of the most flexible and scalable algorithms in this area published to date.

- The stand-alone GPS navigation scenario presents a novel model of the GPS signal error, which correctly models the sequential correlation in the measurements, as well as the horizontal dilution of precision.

- In general, the practical applications presented in this thesis demonstrate that the EnKF is an algorithm that has the merit in robotic applications, not only in the Earth sciences.

Chapter 2

Linear State Estimation

This chapter reviews the state of the art in the area of a linear state estimation, in particular using the most prominent algorithm, the Kalman filter, and several important variants of the algorithm: the Information filter, the Extended Kalman filter (EKF), the Unscented Kalman filter (UKF) and the Ensemble Kalman filter (EnKF). This is a necessary prerequisite for describing the main contribution of this thesis, the Common Past-Invariant Ensemble Kalman filter (CPI-EnKF) and the Augmented EnKF, and for discussion of their features, in particular their computational complexity and treatment of non-linearities. Additionally, the chapter reviews the particle filter, arguably the most widely used probabilistic filter for non-linear non-Gaussian estimation problems, in order to illustrate its principal differences from the EnKF. In this chapter, all the variants of the Kalman filter are defined with the traditional assumption of white independent process and observation noises. The approaches that relax this assumption are discussed in Chapters 3 and 4.

2.1 Linear Discrete-Time Dynamical System Model

The Kalman filter is the most widely used algorithm to estimate the state of a physical system from a sequence of noisy sensor observations. The algorithm is based on a principal assumption that the underlying physical system can be modelled as a linear discrete-time dynamical system, perturbed with an additive Gaussian noise. The system is also assumed to satisfy the Markov property, i.e. the future state of the system depends only upon the present state of the system. An observation of the system represents a linear combination of the state variables, which is also perturbed with an additive Gaussian noise.

The true state of a physical system at a discrete time step t is represented using a column vector $\mathbf{x}_t \in \mathbb{R}^n$,

> *"which is the least amount of data one has to know about the past behavior of the system in order to predict its future behavior."*

as characterised by Rudolf E. Kálmán in his landmark paper (Kalman, 1960). The evolution of the system from a time step $t-1$ to the following time step t is modelled using the following recursive linear equation:

$$\mathbf{x}_t \;=\; \mathbf{F}_t\,\mathbf{x}_{t-1} + \mathbf{B}_t\,\mathbf{u}_t + \mathbf{G}_t\,\mathbf{w}_t \tag{2.1}$$

where $\mathbf{F}_t \in \mathbb{R}^{n\times n}$ is a matrix representing the *process model* (also known as the *state transition model*), $\mathbf{B}_t \in \mathbb{R}^{n\times b}$ is a matrix representing the control-input model applied to the control vector $\mathbf{u}_t \in \mathbb{R}^b$, and $\mathbf{G}_t \in \mathbb{R}^{n\times g}$ is a matrix representing the process noise model applied to the *process noise* term $\mathbf{w}_t \in \mathbb{R}^g$, which is a random vector drawn from a zero-mean (multivariate) Gaussian distribution with a covariance $\mathbf{Q}_t \in \mathbb{R}^{g\times g}$. The function of matrices $\mathbf{F}_t$, $\mathbf{B}_t$ and $\mathbf{G}_t$ is best understood if one considers them as linear mappings on input vectors $\mathbf{x}_{t-1}$, $\mathbf{u}_t$ and $\mathbf{w}_t$, respectively.

The true state of the physical system is hidden and it can only be observed indirectly using noisy sensor measurements. An observation $\mathbf{z}_t \in \mathbb{R}^m$ of the physical system at a time step t is related to the true hidden system state $\mathbf{x}_t$ using the following equation:

$$\mathbf{z}_t \;=\; \mathbf{H}_t\,\mathbf{x}_t + \mathbf{v}_t \tag{2.2}$$

where $\mathbf{H_t} \in \mathbb{R}^{m\times n}$ is a matrix representing the *observation model*, which is essentially a linear mapping from the system state space to the observation space showing what an observation should look like given a specific state vector, and $\mathbf{v}_t \in \mathbb{R}^m$ is the *observation noise*, which is a random vector drawn from a Gaussian distribution with covariance $\mathbf{R}_t$.

For simplicity of the notation, in the rest of this thesis, it will be assumed that the control vector $\mathbf{u}_t$ is always zero, and that the process model $\mathbf{F}_t$, the observation model $\mathbf{H}_t$, the process noise covariance $\mathbf{Q}_t$ and the observation noise covariance $\mathbf{R}_t$ are all constant over time. Hence the time index will be dropped from the respective terms. Furthermore, it will be assumed that the process noise term $\mathbf{w}_t$ has dimension n and its model $\mathbf{G}_t$ is an identity matrix; although this can cause a minor loss of generality in certain formulae, the reader will be referred to the literature containing complete formulae wherever appropriate. All these assumptions lead to a simplified definition of the underlying dynamical system model.

Definition 2.1 (Linear discrete-time dynamical system model). At any time step $t \in \mathbb{N}$, a *linear discrete-time dynamical system* can be fully described by the following equations:

$$\mathbf{x}_t = \mathbf{F}\,\mathbf{x}_{t-1} + \mathbf{w}_t \tag{2.3}$$

$$\mathbf{z}_t = \mathbf{H}\,\mathbf{x}_t + \mathbf{v}_t \tag{2.4}$$

where

$$\begin{aligned}
\mathbf{x}_t &\in \mathbb{R}^n && \text{(true hidden system state at time } t\text{)} \\
\mathbf{F} &\in \mathbb{R}^{n\times n} && \text{(process model)} \\
\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}) &\in \mathbb{R}^n && \text{(process noise from time } t-1 \text{ to } t\text{)} \\
\mathbf{z}_t &\in \mathbb{R}^m && \text{(observation at time } t\text{)} \\
\mathbf{H} &\in \mathbb{R}^{m\times n} && \text{(observation model)} \\
\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}) &\in \mathbb{R}^m && \text{(observation noise at time } t\text{)}
\end{aligned}$$

and the initial state vector $\mathbf{x}_0$ is a random vector drawn from a Gaussian distribution with known parameters:

$$\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$$

Note that $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a (multivariate) Gaussian distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$, of a suitable dimension. Typically, in the literature, the noise term in the process model in Equation (2.3) is denoted $\mathbf{w}_{t-1}$ instead of $\mathbf{w}_t$ (i.e. the time index is shifted by minus one step). However, the notation adopted here better reflects the physical reality - it is the noise term with index t that mostly affects the state estimate at time t. Also, such a notation avoids the need for a special term $\mathbf{w}_0$, and makes many other definitions in the text easier to understand and remember.

Both the process noise sequence $\{\mathbf{w}_t\}$ and the observation noise sequence $\{\mathbf{v}_t\}$ are assumed to be independent Gaussian random processes, also referred to as *white noise*. Furthermore, both the sequences are assumed to be independent of the initial state $\mathbf{x}_0$.

Definition 2.2 (White noise process). An n-dimensional random process $\{\mathbf{w}_t\}$ is called *white noise* if and only if:

(i) For every $t \in \mathbb{N}$, the random vector $\mathbf{w}_t \in \mathbb{R}^n$ has zero mean and a finite covariance $\mathbf{Q}_t \in \mathbb{R}^{n\times n}$, i.e.

$$\mathrm{E}[\mathbf{w}_t] = 0 \tag{2.5}$$

$$\mathrm{cov}(\mathbf{w}_t) = \mathbf{Q}_t \tag{2.6}$$

(ii) The sequence $\{\mathbf{w}_t\}$ is mutually independent, in the sense that any finite subset of its vectors is statistically independent of any other finite subset from the sequence.

Note that the $\mathrm{E}[\mathbf{a}]$ operator denotes the *expected value* (also called the *mean*) of a random vector $\mathbf{a}$, and $\mathrm{cov}(\mathbf{a}, \mathbf{b})$ denotes a cross-covariance between random vectors $\mathbf{a}$ and $\mathbf{b}$, defined as:

$$\mathrm{cov}(\mathbf{a}, \mathbf{b}) = \mathrm{E}\left[(\mathbf{a} - \mathrm{E}[\mathbf{a}])(\mathbf{b} - \mathrm{E}[\mathbf{b}])^{\mathrm{T}}\right] \tag{2.7}$$

Instead of $\mathrm{cov}(\mathbf{a}, \mathbf{a})$, a simplified syntax $\mathrm{cov}(\mathbf{a})$ is used. Definition 2.2 differs from some other definitions of the white noise in the literature, which either only require the process to be sequentially uncorrelated, or assume a constant covariance over time. With the definition adopted here, every white noise process is implied to be sequentially uncorrelated, while the assumption of a constant covariance is not necessary in the context of the Kalman filter.

The assumptions on the properties of the process and observation noise are formalised in the following definition, which is based on (Nieto and Guerrero, 1995).

Definition 2.3 (Independent process and observation noise assumption). Both the process noise sequence $\{\mathbf{w}_t\}$ and the observation noise sequence $\{\mathbf{v}_t\}$, as provided in Definition 2.1, are assumed to have the following properties:

(i) Each of the sequences is white noise, such as that

$$\mathrm{cov}(\mathbf{w}_i, \mathbf{w}_j) = \begin{cases} \mathbf{Q} & i = j \\ 0 & i \neq j \end{cases} \tag{2.8}$$

$$\mathrm{cov}(\mathbf{v}_i, \mathbf{v}_j) = \begin{cases} \mathbf{R} & i = j \\ 0 & i \neq j \end{cases} \tag{2.9}$$

where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and $\mathbf{R} \in \mathbb{R}^{m \times m}$.

(ii) The sequences are mutually independent, therefore

$$\mathrm{cov}(\mathbf{w}_i, \mathbf{v}_j) = 0 \tag{2.10}$$

(iii) Each of the sequences is independent of the initial state, therefore

$$\mathrm{cov}(\mathbf{w}_i, \mathbf{x}_0) = 0 \tag{2.11}$$

$$\mathrm{cov}(\mathbf{v}_i, \mathbf{x}_0) = 0 \tag{2.12}$$

for any $i, j \in \mathbb{N}$.

From a theoretical perspective, the Kalman filter, which will be defined in the following section, assumes that the physical system exactly matches the dynamical system model from Definition 2.1 and that the noises are independent as in Definition 2.3. Although most real-world physical systems do not fit such a model exactly, it turns out that the state of many such systems can be estimated with a reasonable accuracy nevertheless. Random noise in physical processes often appears to have a nearly-Gaussian distribution due to the effects described by the fuzzy central limit theorem, and the effects of non-linearities in the system can be alleviated using various extensions of the Kalman filter.

2.2 Kalman Filter

The Kalman filter, originally introduced in (Kalman, 1960), is a recursive algorithm that computes an estimate of the state of a physical system from a sequence of noisy sensor observations. The goal of the filter is to maintain an estimate of the system state at any time step t, represented using a vector $\hat{\mathbf{x}}_t \in \mathbb{R}^n$, and an estimate of the associated error, represented using a covariance matrix $\mathbf{P}_t \in \mathbb{R}^{n \times n}$, which is *consistent* with the true estimate's error $\mathbf{x}_t - \hat{\mathbf{x}}_t$, i.e.

$$\mathbf{P}_t \; - \; \mathrm{cov}(\mathbf{x}_t - \hat{\mathbf{x}}_t) \; \geq \; 0 \tag{2.13}$$

where $\mathbf{x}_t \in \mathbb{R}^n$ denotes the true system state, and $\mathbf{A} \geq 0$ indicates that a matrix $\mathbf{A}$ is positive-semidefinite. Such a notion of consistency fits the standard definition of consistency (Jazwinski, 2007). An estimate that does not satisfy Equation (2.13) is called an *inconsistent* or *overconfident* estimate. On the other hand, if the estimate is consistent but $\mathbf{P}_t - \mathrm{cov}(\mathbf{x}_t - \hat{\mathbf{x}}_t) \neq 0$, the estimate is called *overpessimistic*, as it assumes a higher uncertainty than necessary.

The Kalman filter operates recursively in two logical steps: the *prediction* step and the *update* step. In the prediction step, the state estimate $\hat{\mathbf{x}}_{t-1}$ and associated error covariance $\mathbf{P}_{t-1}$ at a time step $t-1$ is used to predict the *a priori* state estimate $\hat{\mathbf{x}}_t^-$ and the associated error covariance $\mathbf{P}_t^-$ at the following time step t. The prediction step effectively transforms the state estimate using the process model, while increasing the associated uncertainty to account for the effects of the process noise based on the model of the system. Note that $\hat{\mathbf{x}}_{t-1}$ and $\mathbf{P}_{t-1}$ represent all the information necessary to compute $\hat{\mathbf{x}}_t^-$ and $\mathbf{P}_t^-$, due to the Markov property of the dynamical system. In the update step, the *a priori* state estimate $\hat{\mathbf{x}}_t^-$ and covariance $\mathbf{P}_t^-$ is refined using a physical observation of the system $\mathbf{z}_t$ at a time step t, in order to provide the *a posteriori* estimate $\hat{\mathbf{x}}_t$ and covariance $\mathbf{P}_t$, which will generally have a lower uncertainty than the

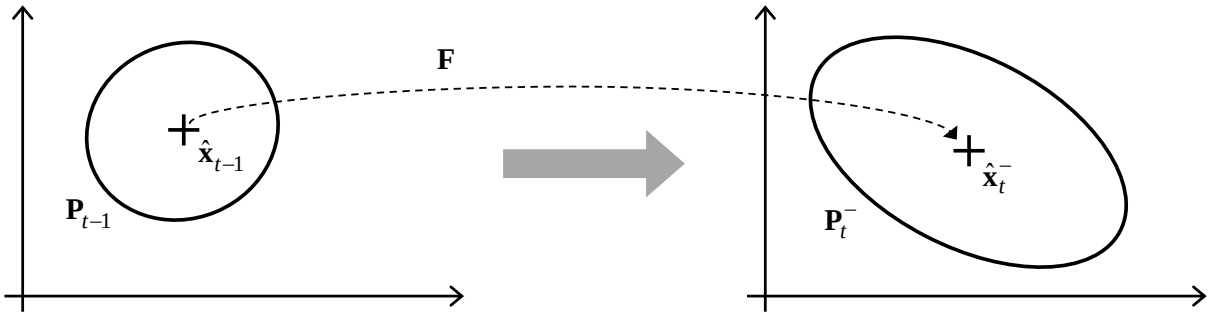


Fig. 2.1: Prediction step of the Kalman filter.

a priori estimate. Illustrations of the prediction step and the update step of the Kalman filter in a dynamical system with $n = 2$ and $m = 1$ are provided in Figures 2.1 and 2.2, respectively.

Definition 2.4 (Kalman filter). Suppose a physical system is modelled as a linear discrete-time dynamical system from Definition 2.1, with an independent process and observation noise as in Definition 2.3. The estimate $\hat{\mathbf{x}}_t \in \mathbb{R}^n$ of the system state at a time $t \in \mathbb{N}$, given all the observations $\mathbf{z}_1, \ldots, \mathbf{z}_t$ up to time t, and the estimate's error covariance matrix $\mathbf{P}_t \in \mathbb{R}^{n \times n}$, can be computed recursively from the previous state estimate $\hat{\mathbf{x}}_{t-1} \in \mathbb{R}^n$ and the associated error covariance $\mathbf{P}_{t-1} \in \mathbb{R}^{n \times n}$ using the following equations:

(i) Prediction step

$$\hat{\mathbf{x}}_t^- = \mathbf{F}\,\hat{\mathbf{x}}_{t-1} \qquad (\textit{a priori} \text{ state estimate}) \tag{2.14}$$

$$\mathbf{P}_t^- = \mathbf{F}\,\mathbf{P}_{t-1}\,\mathbf{F}^{\mathrm{T}} + \mathbf{Q} \qquad (\textit{a priori} \text{ estimate error covariance}) \tag{2.15}$$

(ii) Update step

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t\left(\mathbf{z}_t - \mathbf{H}\,\hat{\mathbf{x}}_t^-\right) \qquad (\textit{a posteriori} \text{ state estimate}) \tag{2.16}$$

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t\,\mathbf{H}\,\mathbf{P}_t^- \qquad (\textit{a posteriori} \text{ estimate error covariance}) \tag{2.17}$$

$$\mathbf{K}_t = \mathbf{P}_t^-\,\mathbf{H}^{\mathrm{T}}\left[\,\mathbf{H}\,\mathbf{P}_t^-\,\mathbf{H}^{\mathrm{T}} + \mathbf{R}\,\right]^{-1} \qquad (\text{optimal Kalman gain factor}) \tag{2.18}$$

The initial state estimate $\hat{\mathbf{x}}_0$ and the error covariance $\mathbf{P}_0$ are assumed to be equal to the parameters of the Gaussian probability distribution of the initial state, as described in Definition 2.1. By convention and without a loss of generality, it is assumed that no observation is available at time $t = 0$.

Note that the term $\mathbf{H}\,\hat{\mathbf{x}}_t^-$ computes the *expected observation* given the current state estimate. Hence, the term $\mathbf{z}_t - \mathbf{H}\,\hat{\mathbf{x}}_t^-$ is often referred to as the *innovation* or the *measurement residual*, whose covariance is estimated by the term $\mathbf{H}\mathbf{P}_t^-\mathbf{H}^{\mathrm{T}} + \mathbf{R}$.

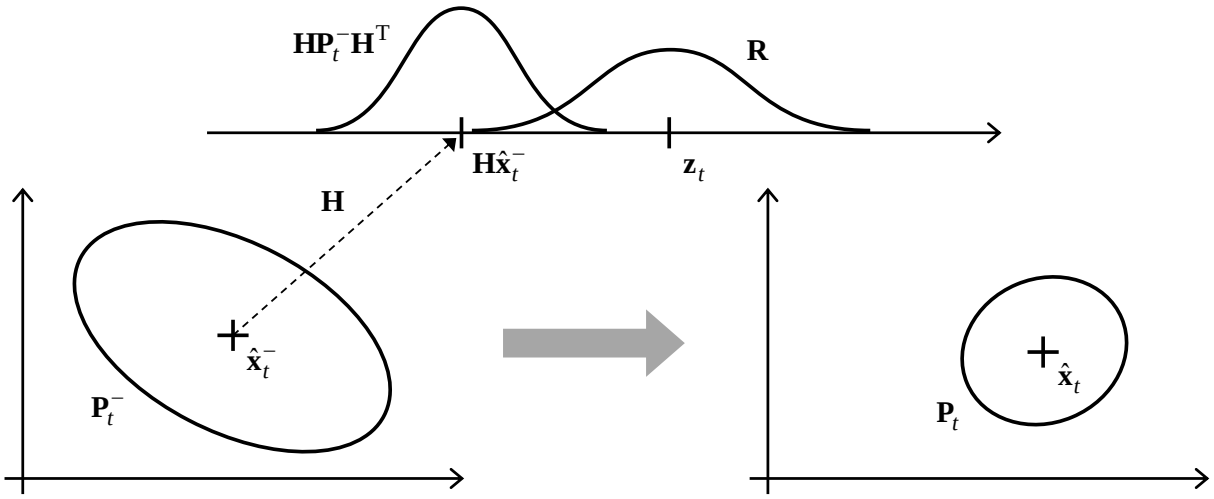


Fig. 2.2: Update step of the Kalman filter.

If all the assumptions of the Kalman filter are satisfied, all the probability distributions involved are Gaussian and the dynamical system is modelled precisely, then the resulting state estimate is guaranteed to have a minimum possible mean squared error with respect to the true system state, and the estimate of the error covariance is exact, i.e.

$$\mathbf{P}_t \;=\; \mathrm{cov}(\mathbf{x}_t - \hat{\mathbf{x}}_t) \tag{2.19}$$

A formal proof of this statement and additional details can be found in numerous publications on the Kalman filter, such as the original (Kalman, 1960) paper, or books such as (Gelb, 1974; Simon, 2006; Grewal and Andrews, 2011; Brown and Hwang, 2012). A variant of the Kalman filter based on the more general linear dynamical system model as in Equation (2.1) can be found in (Maybeck, 1979). Note that the update equation is effectively a weighted average of the state estimate and the observation, with the weight given by the *Kalman gain* factor $\mathbf{K}_t$, which depends on the covariances of the *a priori* estimate and the observation. The higher the Kalman gain, the more information from the observation makes its way to the *a posteriori* state estimate, and vice versa.

Due to the recursive nature of the Kalman filter algorithm, only the current observation, state estimate and error covariance need to be preserved at any time; no other information is necessary. This feature stems from the Markov property of the dynamical system, its linearity, the Gaussian nature of the probability distributions involved, and the fact that a product and a convolution of two Gaussian distributions is also a Gaussian distribution (Bromiley, 2003). As such, the computational costs of the algorithm do not change over time, and hence it can be

	Time	Space
Prediction step	$O(\, n^3 \,)$	$O(\, n^2 \,)$
Update step	$O(\, n^2 m + n m^2 + m^3 \,)$	$O(\, n^2 + n m + m^2 \,)$

Table 2.1: Asymptotic computational complexity of the Kalman filter

applied in real-time estimation problems. The asymptotic computational complexity of a direct implementation of the Kalman filter based on Definition 2.4 is provided in Table 2.1, where n and m denote the number of dimensions of the state space and the observation, respectively. Note that the most tedious operation - the prediction step with $O(n^3)$ time complexity - can be performed more efficiently in problems where only a part of the state vector is altered by the process model. For example, in *simultaneous localisation and mapping* (SLAM), the robot movement only updates a small portion of the map, and hence the prediction step can be implemented in $O(n)$ time (Smith et al., 1990). Similarly, in certain estimation problems, it is possible to exploit the structure of the state space in order to reduce the computational cost of the update step, in particular to eliminate the $O(n^2)$ term. For example, in SLAM it is possible to decompose the state estimate's vector and covariance matrix (representing a global map) into several smaller components (local sub-maps) and maintain these separately. The component estimates only need to be combined sporadically and thus the quadratic computation cost can be amortised between more steps of the filter (Paz et al., 2007; Huang et al., 2008). However, these approaches are beyond the scope of this thesis, as it deals with general estimation problems.

The Kalman filter's favourable computational properties and general simplicity of implementation, its versatility to various problems and observation schedules, and the fact that it often provides reasonable state estimates even if the model of the dynamical system is not precise, are arguably the main reasons why it reached such popularity and widespread adoption in technology over the last 50 years. The applications of the Kalman filter span areas such as guidance, navigation and control, robotics, target tracking, or time series analysis in signal processing and econometrics. As Harold W. Sorenson famously put it (Sorenson, 1985),

> *"the Kalman filter represents the most widely applied and demonstrably useful result to emerge from the state variable approach of modern control theory."*

From the perspective of statistics, the Kalman filter can be viewed as a special case of a recursive Bayesian inference algorithm. Specifically, the update step of the Kalman filter

represents an application of Bayes' rule, in which the probability estimate of a hypothesis is updated as more evidence becomes available, formally:

$$P(\mathit{Hypothesis} \,|\, \mathit{Evidence}) \;\propto\; P(\mathit{Evidence} \,|\, \mathit{Hypothesis})\, P(\mathit{Hypothesis}) \tag{2.20}$$

In the context of the linear discrete-time dynamical system model from Definition 2.1, the state estimate at a time t given all the observations $\mathbf{z}_1, \ldots, \mathbf{z}_t$ up to time t can be denoted as $P(\mathbf{x}_t \,|\, \mathbf{z}_1, \ldots, \mathbf{z}_t)$ and expressed using Bayes' rule as:

$$P(\mathbf{x}_t \,|\, \mathbf{z}_1, \ldots, \mathbf{z}_t) \;\propto\; P(\mathbf{z}_t \,|\, \mathbf{x}_t, \mathbf{z}_1, \ldots, \mathbf{z}_{t-1})\, P(\mathbf{x}_t \,|\, \mathbf{z}_1, \ldots, \mathbf{z}_{t-1}) \tag{2.21}$$

This expression can be rewritten due to Markov assumption to:

$$P(\mathbf{x}_t \,|\, \mathbf{z}_1, \ldots, \mathbf{z}_t) \;\propto\; P(\mathbf{z}_t \,|\, \mathbf{x}_t)\; P(\mathbf{x}_t \,|\, \mathbf{z}_1, \ldots, \mathbf{z}_{t-1}) \tag{2.22}$$

and the application of the law of total probability then leads to the following expression:

$$P(\mathbf{x}_t \,|\, \mathbf{z}_1, \ldots, \mathbf{z}_t) \;\propto\; P(\mathbf{z}_t \,|\, \mathbf{x}_t) \int_{\mathbf{x}_{t-1}} P(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, \mathbf{z}_1, \ldots, \mathbf{z}_{t-1})\; P(\mathbf{x}_{t-1} \,|\, \mathbf{z}_1, \ldots, \mathbf{z}_{t-1})\, \mathrm{d}\mathbf{x}_{t-1} \tag{2.23}$$

Finally, this expression can be rewritten due to Markov property to the recursive Bayesian inference formulation:

$$P(\mathbf{x}_t \,|\, \mathbf{z}_1, \ldots, \mathbf{z}_t) \;\propto\; P(\mathbf{z}_t \,|\, \mathbf{x}_t) \int_{\mathbf{x}_{t-1}} P(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})\; P(\mathbf{x}_{t-1} \,|\, \mathbf{z}_1, \ldots, \mathbf{z}_{t-1})\, \mathrm{d}\mathbf{x}_{t-1} \tag{2.24}$$

Note that the Kalman filter is just a special case of this recursive Bayesian inference algorithm, in which the probability distributions are all Gaussian and the dynamical system models linear. The prediction step of the Kalman filter effectively facilitates the integration in Equation (2.24) while the update step facilitates the multiplication of the integral by the term from its left side. A detailed analysis of the Kalman filter in the context of Bayesian inference can be found in (Meinhold and Singpurwalla, 1983).

The Kalman filter, in its simple form described in this section, has three key limitations. First, the process model and the observation model are assumed linear not only from a theoretical perspective, but also from the perspective of a practical implementation; both the prediction and update step require the models to be expressed as matrices $\mathbf{F}$ and $\mathbf{H}$, respectively. Second, the Kalman filter does not scale and it cannot be applied to problems with a large number of state-space or observation dimensions. Both these limitations are addressed by other variants of the Kalman filter, which will be reviewed in the following sections. Third, the simple Kalman filter makes some strong assumptions on the process and observation noises, in particular that they are white and independent. Relaxation of these noise assumptions represents the core problem addressed in this thesis.

2.3 Information Filter

The Information filter, originally proposed in (Fraser, 1967), is an algebraic re-formulation of the Kalman filter, in which the uncertainty associated with a state estimate is not represented using a covariance matrix, but rather using its inverse - the so-called *information matrix*. The Information filter is based on exactly the same system model and assumptions as the Kalman filter, and they both provide algebraically equivalent estimates. From the perspective of this thesis, the Information filter is important because it is a building block of several decentralised data fusion approaches, which will be described in Chapter 4.

Formally, in the Information filter, a state estimate $\hat{\mathbf{x}}_t \in \mathbb{R}^n$ and an associated error covariance $\mathbf{P}_t \in \mathbb{R}^{n \times n}$ are represented using an *information vector* $\hat{\mathbf{y}}_t \in \mathbb{R}^n$ and an *information matrix* $\mathbf{Y}_t \in \mathbb{R}^{n \times n}$, respectively, defined as:

$$\hat{\mathbf{y}}_t = \mathbf{P}_t^{-1} \hat{\mathbf{x}}_t \tag{2.25}$$

$$\mathbf{Y}_t = \mathbf{P}_t^{-1} \tag{2.26}$$

Similarly to the Kalman filter, the Information filter defines recursive equations to compute the information vector and matrix, and these equations are also logically divided into the prediction and the update step.

Definition 2.5 (Information filter). Suppose a physical system is modelled as a linear discrete-time dynamical system from Definition 2.1, with independent zero-mean process and observation noises as in Definition 2.3. The estimate of the system state at a time $t \in \mathbb{N}$, given all the observations $\mathbf{z}_1, \ldots, \mathbf{z}_t$ up to time t, represented using an information vector $\hat{\mathbf{y}}_t \in \mathbb{R}^n$ and an information matrix $\mathbf{Y}_t \in \mathbb{R}^{n \times n}$, can be computed recursively from the previous state estimate, represented by information vector $\hat{\mathbf{y}}_{t-1} \in \mathbb{R}^n$ and information matrix $\mathbf{Y}_{t-1} \in \mathbb{R}^{n \times n}$, using the following equations:

(i) Prediction step

$$\hat{\mathbf{y}}_t^- = (I - \mathbf{C}_t) (\mathbf{F}^{-1})^{\mathrm{T}} \hat{\mathbf{y}}_{t-1} \qquad (\textit{a priori} \text{ information vector}) \tag{2.27}$$

$$\mathbf{Y}_t^- = (I - \mathbf{C}_t) \mathbf{M}_t (I - \mathbf{C}_t)^{\mathrm{T}} + \mathbf{C}_t \mathbf{Q}^{-1} \mathbf{C}_t^{\mathrm{T}} \qquad (\textit{a priori} \text{ information matrix}) \tag{2.28}$$

$$\mathbf{C}_t = \mathbf{M}_t \left[\mathbf{M}_t + \mathbf{Q}^{-1} \right]^{-1} \tag{2.29}$$

$$\mathbf{M}_t = (\mathbf{F}^{-1})^{\mathrm{T}} \mathbf{Y}_{t-1} \mathbf{F}^{-1} \tag{2.30}$$

(ii) Update step

$$\hat{\mathbf{y}}_t \;=\; \hat{\mathbf{y}}_t^- + \mathbf{H}^{\mathrm{T}}\,\mathbf{R}^{-1}\,\mathbf{z}_t \qquad (a\ \textit{posteriori}\ \text{information vector}) \tag{2.31}$$

$$\mathbf{Y}_t \;=\; \mathbf{Y}_t^- + \mathbf{H}^{\mathrm{T}}\,\mathbf{R}^{-1}\,\mathbf{H} \qquad (a\ \textit{posteriori}\ \text{information matrix}) \tag{2.32}$$

The remarks from Definition 2.4 on the properties of the initial state estimate apply here too. Note that I above denotes identity matrix of an appropriate dimension.

The equations of the Information filter described above can be derived directly from the equations of the Kalman filter, using the matrix inversion lemma or by other means, as shown in (Brown and Hwang, 2012) or (Anderson and Moore, 1979). The most remarkable feature of the Information filter is the simplicity of the update step, which enables the filter to assimilate an observation into the state estimate by addition. If there are multiple observations of the physical system at a time, they all can be added in the same way. Formally, suppose that at a time step t there are k independent observations $\mathbf{z}_{i|t} \in \mathbb{R}^{m_i}$ for $i = 1, \ldots, k$, each of them modelled as:

$$\mathbf{z}_{i|t} \;=\; \mathbf{H}_i\,\mathbf{x}_t \;+\; \mathbf{v}_{i|t} \tag{2.33}$$

where $\mathbf{H}_i \in \mathbb{R}^{m_i \times n}$ denotes an observation model, and $\mathbf{v}_{i|t} \in \mathbb{R}^{m_i}$ denotes an observation noise term with a covariance $\mathbf{R}_i = \mathrm{cov}(\mathbf{v}_{i|t}) \in \mathbb{R}^{m_i \times m_i}$. The update step of the Information filter then has the following form:

$$\hat{\mathbf{y}}_t \;=\; \hat{\mathbf{y}}_t^- + \sum_{i=1}^{k} \mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{z}_{i|t} \tag{2.34}$$

$$\mathbf{Y}_t \;=\; \mathbf{Y}_t^- + \sum_{i=1}^{k} \mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{H}_i \tag{2.35}$$

Note that with the traditional form of the Kalman filter update equations from Definition 2.4, such a summation is generally not possible (Durrant-Whyte et al., 2001), i.e.

$$\hat{\mathbf{x}}_t \;\neq\; \hat{\mathbf{x}}_t^- + \sum_{i=1}^{k} \mathbf{K}_{i|t}\left(\mathbf{z}_{i|t} \;-\; \mathbf{H}_i\,\hat{\mathbf{x}}_t^-\right) \tag{2.36}$$

where $\mathbf{K}_{i|t}$ denotes the Kalman gain for the respective observations.

The simplicity of the update step in the Information filter comes at the cost of higher complexity of the prediction step, which, in fact, is a dual to the update step of the traditional Kalman filter. Therefore, many practical applications employ the prediction step of the traditional Kalman filter and the update step of the Information filter, and they convert between

	Time	Space
Prediction step	$O(\, n^3 \,)$	$O(\, n^2 \,)$
Update step	$O(\, n^2 m + n m^2 \,)$	$O(\, n^2 + n m + m^2 \,)$

Table 2.2: Asymptotic computational complexity of the Information filter

$(\hat{\mathbf{x}}_t, \mathbf{P}_t)$ and $(\hat{\mathbf{y}}_t, \mathbf{Y}_t)$, and vice versa, as necessary. Such prediction step equations then look as follows:

$$\hat{\mathbf{y}}_t^- = \mathbf{Y}_t^- \, \mathbf{F} \, \mathbf{Y}_{t-1}^{-1} \, \hat{\mathbf{y}}_{t-1} \tag{2.37}$$

$$\mathbf{Y}_t^- = \left[\mathbf{F} \, \mathbf{Y}_{t-1}^{-1} \, \mathbf{F}^{\mathrm{T}} + \mathbf{Q} \right]^{-1} \tag{2.38}$$

Another advantage of the Information filter compared to the Kalman filter is the fact that the information matrix can, unlike the covariance matrix, represent an infinite uncertainty, which is especially useful for the initial state estimate. Therefore, in some applications, the first step of the Kalman filter is performed in its information form to guarantee formal correctness of the estimates (Rogers, 1987).

The asymptotic computational complexity of a direct implementation of the Information filter equations from Definition 2.5, assuming that $\mathbf{F}^{-1}$, $\mathbf{Q}^{-1}$ and $\mathbf{R}^{-1}$ are time-invariant and hence pre-computed, is summarised in Table 2.2. Although the Information filter is asymptoticly faster than the Kalman filter by avoiding the $O(m^3)$ computation in the update step, the additional conversion of an n-dimensional information vector and matrix to a standard vector and covariance matrix, which is often necessary for a practical interpretation of the filter's estimates, has an additional asymptotic time complexity $O(n^3)$.

2.4 Extended Kalman Filter

The Extended Kalman filter (EKF) is a straightforward generalisation of the Kalman filter that allows it to operate with non-linear process and observation models. The models no longer need to be represented as a matrix, instead they can be represented as a general differentiable function. The EKF was developed by NASA soon after the introduction of the Kalman filter, when analysing possibilities of its application in space flight, in particular for navigation of vehicles during lunar missions (Smith et al., 1962; McElhoe, 1966). A formal definition of the underlying non-linear dynamical system model, and the EKF, is provided as follows:

Definition 2.6 (Non-linear discrete-time dynamical system model). At any time step $t \in \mathbb{N}$, a *non-linear discrete-time dynamical system* is fully described by the following equations:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_t \tag{2.39}$$

$$\mathbf{z}_t = h(\mathbf{x}_t) + \mathbf{v}_t \tag{2.40}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}^n$ is the process model, $h\colon \mathbb{R}^n \to \mathbb{R}^m$ is the observation model, and the other terms are as in Definition 2.1.

Definition 2.7 (Extended Kalman filter). Suppose a physical system is modelled as a non-linear discrete-time dynamical system from Definition 2.6, with independent zero-mean process and observation noise as in Definition 2.3. The estimate $\hat{\mathbf{x}}_t \in \mathbb{R}^n$ of the system state at a time $t \in \mathbb{N}$, given all the observations $\mathbf{z}_1, \ldots, \mathbf{z}_t$ up to time t, and the covariance matrix $\mathbf{P}_t \in \mathbb{R}^{n \times n}$ of the estimate's error, can be computed recursively from the previous state estimate $\hat{\mathbf{x}}_{t-1} \in \mathbb{R}^n$ and the associated error covariance $\mathbf{P}_{t-1} \in \mathbb{R}^{n \times n}$ using the following equations:

(i) Prediction step

$$\hat{\mathbf{x}}_t^- = f(\hat{\mathbf{x}}_{t-1}) \qquad (a\ \textit{priori}\ \text{state estimate}) \tag{2.41}$$

$$\mathbf{P}_t^- = \mathcal{F}_t \mathbf{P}_{t-1} \mathcal{F}_t^{\mathrm{T}} + \mathbf{Q} \qquad (a\ \textit{priori}\ \text{estimate error covariance}) \tag{2.42}$$

$$\mathcal{F}_t = J_f(\hat{\mathbf{x}}_{t-1}) \qquad (\text{Jacobian of } f \text{ evaluated at } \hat{\mathbf{x}}_{t-1}) \tag{2.43}$$

(ii) Update step

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t \left(\mathbf{z}_t - h(\hat{\mathbf{x}}_t^-)\right) \qquad (a\ \textit{posteriori}\ \text{state estimate}) \tag{2.44}$$

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t \mathcal{H}_t \mathbf{P}_t^- \qquad (a\ \textit{posteriori}\ \text{estimate error covariance}) \tag{2.45}$$

$$\mathbf{K}_t = \mathbf{P}_t^- \mathcal{H}_t^{\mathrm{T}} \left[\mathcal{H}_t \mathbf{P}_t^- \mathcal{H}_t^{\mathrm{T}} + \mathbf{R}\right]^{-1} \qquad (\text{Kalman gain factor}) \tag{2.46}$$

$$\mathcal{H}_t = J_h(\hat{\mathbf{x}}_t^-) \qquad (\text{Jacobian of } h \text{ evaluated at } \hat{\mathbf{x}}_t^-) \tag{2.47}$$

The remarks from Definition 2.4 on the properties of the initial state estimate $\hat{\mathbf{x}}_0$ and the covariance $\mathbf{P}_0$ apply here too.

Recall that the Jacobian of a function $g\colon \mathbb{R}^n \to \mathbb{R}^m$, which is given by m real-valued component functions $g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)$, is a function $J_g\colon \mathbb{R}^n \to \mathbb{R}^{m \times n}$ defined as:

$$J_g = \begin{bmatrix} \dfrac{\partial g_1}{\partial x_1} & \cdots & \dfrac{\partial g_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial g_m}{\partial x_1} & \cdots & \dfrac{\partial g_m}{\partial x_n} \end{bmatrix} \tag{2.48}$$

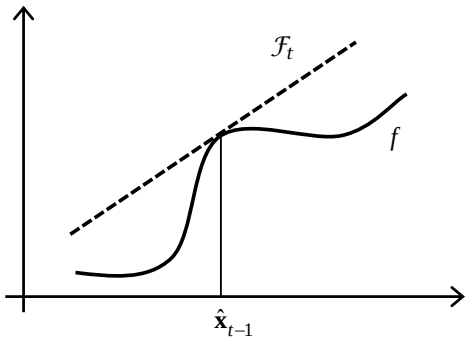


Fig. 2.3: Linearisation of the prediction model in the Extended Kalman filter.

The $n \times m$ components of the Jacobian correspond to all n first-order partial derivatives of all m component functions of g. When the Jacobian function J_g is evaluated at some point $(x_1, \ldots, x_n)$, the resulting matrix represents the slope of the tangent line in each of the n axes to each of the m component functions. Such a matrix defines a linear mapping, which effectively approximates the function g around the point $(x_1, \ldots, x_n)$. An example of such a *linearisation* of a non-linear prediction model in a one-dimensional dynamical system is depicted in Figure 2.3.

Note that if both the process model f and the observation model h are linear mappings, i.e. they have the property that:

$$f(a_1\mathbf{x}_1 + \cdots + a_k\mathbf{x}_k) = a_1 f(\mathbf{x}_1) + \cdots + a_k f(\mathbf{x}_k) \tag{2.49}$$

$$h(a_1\mathbf{x}_1 + \cdots + a_k\mathbf{x}_k) = a_1 h(\mathbf{x}_1) + \cdots + a_k h(\mathbf{x}_k) \tag{2.50}$$

for any vectors $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{R}^n$ and scalars $a_1, \ldots, a_k \in \mathbb{R}$, then the EKF reduces to the simple Kalman filter because the Jacobians $\mathcal{F}_t$ and $\mathcal{H}_t$ from Definition 2.7 will be constant over time and equal to the matrices $\mathbf{F}$ and $\mathbf{H}$ from Definition 2.4, which furthermore correspond to the matrices defining the linear mappings f and h, respectively. Therefore, in the case of a linear process and observation model, the EKF also computes minimum mean squared error state estimates and consistent covariances, given the same assumptions as the simple Kalman filter.

However, the EKF is not an optimal estimator in general. With non-linear process or observation models, the Jacobians merely provide a linear approximation of these models, and therefore the estimates are also only approximate. The errors stemming from such a linearisation may cause the filter to provide overconfident estimates and to diverge. Another issue with the EKF is the need to evaluate the Jacobians of the process and observation models, which might be a difficult task per se. Partial derivatives of certain functions do not have a

closed form, and therefore numerical approximations need to be used, which further increases the computational cost. The EKF is well characterised in (Julier and Uhlmann, 2004):

> *"The extended Kalman filter (EKF) is probably the most widely used estimation algorithm for nonlinear systems. However, more than 35 years of experience in the estimation community has shown that* [it] *is difficult to implement, difficult to tune, and only reliable for systems that are almost linear on the time scale of the updates."*

The problems of the overconfidence and divergence in the EKF can sometimes be alleviated by introducing an additional stabilising noise in the system model, but such a measure is suboptimal as it effectively discards potentially useful information and slows the convergence rate. Alternatively, the estimation error can be reduced using iteration, leading to the *Iterated Extended Kalman filter* (Jazwinski, 2007), which, however, further increases the computational cost of the filter. The limitations of the EKF motivated the development of other variants of the Kalman filter, which are described in the following sections.

Despite all its problems, the EKF has been applied to a wide range of non-linear estimation problems, for example, in navigation and global positioning systems (Kaplan and Hegarty, 2006). Assuming the process model f and the observation model h can be evaluated in a constant time, and their Jacobians are available analytically, the asymptotic computational complexity of the EKF is equivalent to that of the simple Kalman filter, as described in Table 2.1.

2.5 Unscented Kalman Filter

The *Unscented Kalman filter* (UKF), introduced in (Julier and Uhlmann, 1996, 1997b), is a variant of the Kalman filter that can operate with non-linear process and observation models. However, compared to the EKF, the UKF does not require the Jacobians of the models, and it provides more accurate state and covariance estimates, in particular in the case of highly non-linear models (Julier and Uhlmann, 2004).

The UKF is based on the *unscented transformation* (UT), a general method that can propagate the mean and the covariance of a probability distribution through a non-linear function. The main idea behind the UT is that it is easier to approximate a probability distribution, than to approximate a non-linear function. The probability distribution is approximated using a set of deterministically selected weighted samples, the so-called *sigma points*. The sigma points are defined in such a way that their weighted average and weighted sample covariance are equal to

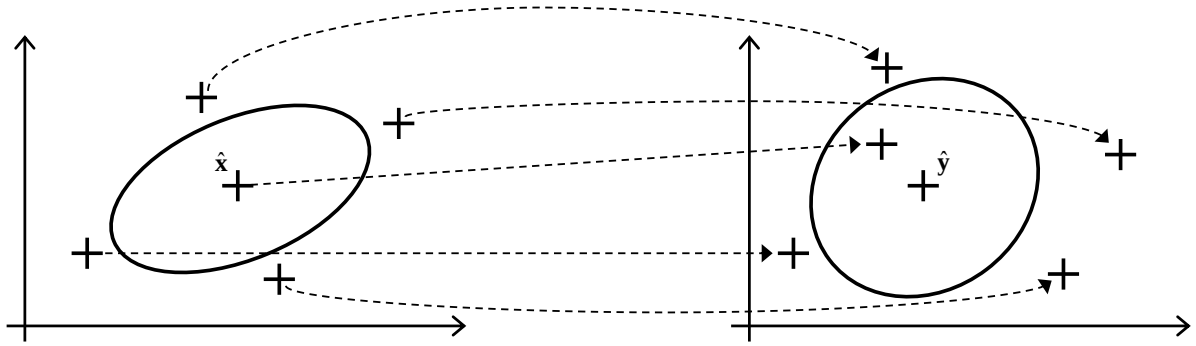


Fig. 2.4: Unscented transformation.

the mean and the covariance of the original distribution, respectively. If the sigma points are propagated through a non-linear function, they can be used to recover the mean and the covariance of the propagated probability distribution. An example of an unscented transformation in two-dimensional spaces is depicted in Figure 2.4.

Definition 2.8 (Unscented transformation). Let $\mathbf{x} \in \mathbb{R}^n$ be a random vector that has a probability distribution with mean $\hat{\mathbf{x}} \in \mathbb{R}^n$ and covariance $\mathbf{P_{xx}} \in \mathbb{R}^{n \times n}$. If the random vector $\mathbf{x}$ is transformed using a function $g \colon \mathbb{R}^n \to \mathbb{R}^m$, then the mean $\hat{\mathbf{y}} \in \mathbb{R}^m$ and the covariance $\mathbf{P_{yy}} \in \mathbb{R}^{m \times m}$ of the probability distribution of the transformed random vector $\mathbf{y} = g(\mathbf{x})$ can be approximated using the following steps:

(i) Compute the sigma points $\mathcal{X}_0, \ldots, \mathcal{X}_{2n} \in \mathbb{R}^n$ as:

$$\begin{aligned} \mathcal{X}_0 &= \hat{\mathbf{x}} \\ \mathcal{X}_i &= \hat{\mathbf{x}} + \left(\sqrt{(n+\kappa)\mathbf{P_{xx}}}\right)_i \\ \mathcal{X}_{n+i} &= \hat{\mathbf{x}} - \left(\sqrt{(n+\kappa)\mathbf{P_{xx}}}\right)_i \end{aligned} \tag{2.51}$$

and the corresponding weight coefficients $W_0, \ldots, W_{2n} \in \mathbb{R}$ as:

$$\begin{aligned} W_0 &= \frac{\kappa}{n+\kappa} \\ W_i &= \frac{1}{2(n+\kappa)} \\ W_{n+i} &= \frac{1}{2(n+\kappa)} \end{aligned} \tag{2.52}$$

for $i = 1, \ldots, n$, where $\kappa \in \mathbb{R}$ is a free parameter, and $\left(\sqrt{\mathbf{A}}\right)_i$ denotes the ith row of the square root of a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, i.e. a matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ such as $\mathbf{B}\mathbf{B}^{\mathrm{T}} = \mathbf{A}$.

(ii) Compute the transformed sigma points $\mathcal{Y}_0, \ldots, \mathcal{Y}_{2n} \in \mathbb{R}^m$ as:

$$\mathcal{Y}_i \,=\, g(\,\mathcal{X}_i\,) \tag{2.53}$$

for $i = 0, \ldots, 2n$.

(iii) The mean $\hat{\mathbf{y}}$ and the covariance $\mathbf{P_{yy}}$ of the probability distribution of the transformed random vector $\mathbf{y} = g(\mathbf{x})$ is approximated as:

$$\hat{\mathbf{y}} \,\approx\, \sum_{i=0}^{2n} W_i\, \mathcal{Y}_i \tag{2.54}$$

$$\mathbf{P_{yy}} \,\approx\, \sum_{i=0}^{2n} W_i\, (\mathcal{Y}_i - \hat{\mathbf{y}})(\mathcal{Y}_i - \hat{\mathbf{y}})^{\mathrm{T}} \tag{2.55}$$

Note that the sigma points $\mathcal{X}_0, \ldots, \mathcal{X}_{2n}$ and weights $W_0, \ldots, W_{2n}$ are defined in such a way that their weighted average and weighted sample covariance equal the original mean and covariance of $\mathbf{x}$, respectively, i.e.

$$\hat{\mathbf{x}} \,=\, \sum_{i=0}^{2n} W_i\, \mathcal{X}_i \tag{2.56}$$

$$\mathbf{P_{xx}} \,=\, \sum_{i=0}^{2n} W_i\, (\mathcal{X}_i - \hat{\mathbf{x}})(\mathcal{X}_i - \hat{\mathbf{x}})^{\mathrm{T}} \tag{2.57}$$

and the weights have a property that $\sum_{i=0}^{2n} W_i = 1$. Additionally, the cross-covariance $\mathbf{P_{xy}} = \mathrm{cov}(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n \times m}$ between random vectors $\mathbf{x}$ and $\mathbf{y}$ can be approximated as:

$$\mathbf{P_{xy}} \,\approx\, \sum_{i=0}^{2n} W_i\, (\mathcal{X}_i - \hat{\mathbf{x}})(\mathcal{Y}_i - \hat{\mathbf{y}})^{\mathrm{T}} \tag{2.58}$$

In the following text, such an unscented transformation will be denoted using the following notation:

$$(\,\hat{\mathbf{x}},\, \mathbf{P_{xx}}\,) \xrightarrow{\;g\;} (\,\hat{\mathbf{y}},\, \mathbf{P_{yy}}\,) \tag{2.59}$$

The value of the parameter κ in Definition 2.8 affects how well particular types of distributions can be approximated, in particular their third and higher moments (Van Der Merwe et al., 2000). Note that alternative sigma point weighting schemes have been proposed in the literature, which might also facilitate more accurate approximations in certain situations. Recommendations on the parameter and weighting scheme selection, as well as other properties of the unscented transformation, are discussed in detail in (Wan and Van Der Merwe, 2000; Julier and Uhlmann, 2004).

The unscented transformation in Definition 2.8 has a relatively high asymptotic time complexity $O(n^3+nm^2)$, due to the costly computation of the square root of the covariance matrix in Equation (2.51), typically performed using the Cholesky decomposition (Press et al., 2007), and due to the computation of the sample covariance in Equation (2.55). The optional computation of the cross-covariance in Equation (2.58) has an additional time complexity of $O(n^2m)$.

The Kalman filter performs a potentially non-linear transformation of a random vector on two different occasions: when computing the *a priori* state estimate using the process model in the prediction step, and when computing the expected observation from the *a priori* state estimate using the observation model in the update step. If both these transformations are replaced by the unscented transformation, the resulting filter is called the Unscented Kalman filter (UKF). However, in order to apply the unscented transformation, the filter equations need to be reformulated in such a way that the transformation functions also perform the addition of the process and observation noises. This requires the state estimate and its error covariance to be augmented with the mean and the covariance of the respective noises.

Definition 2.9 (Unscented Kalman filter). Suppose a physical system is modelled as a non-linear discrete-time dynamical system from Definition 2.6, with independent zero-mean process and observation noise as in Definition 2.3. The estimate $\hat{\mathbf{x}}_t \in \mathbb{R}^n$ of the system state at a time $t \in \mathbb{N}$, given all the observations $\mathbf{z}_1, \ldots, \mathbf{z}_t$ up to time t, and the covariance matrix $\mathbf{P}_t \in \mathbb{R}^{n\times n}$ of the estimate's error, can be computed recursively from the previous state estimate $\hat{\mathbf{x}}_{t-1} \in \mathbb{R}^n$ and associated covariance $\mathbf{P}_{t-1} \in \mathbb{R}^{n\times n}$ as follows:

(i) **Prediction step**

Define the augmented state estimate $\hat{\mathbf{x}}^*_{t-1} \in \mathbb{R}^{2n}$ and the augmented error covariance $\mathbf{P}^*_{t-1} \in \mathbb{R}^{2n\times 2n}$ as:

$$\hat{\mathbf{x}}^*_{t-1} = \begin{bmatrix} \hat{\mathbf{x}}_{t-1} \\ 0 \end{bmatrix} \qquad \mathbf{P}^*_{t-1} = \begin{bmatrix} \mathbf{P}_{t-1} & 0 \\ 0 & \mathbf{Q} \end{bmatrix} \tag{2.60}$$

and the augmented process model $f^*\colon \mathbb{R}^{2n} \to \mathbb{R}^n$ as:

$$f^*(\mathbf{x}, \mathbf{w}) = f(\mathbf{x}) + \mathbf{w} \tag{2.61}$$

where $\mathbf{x}, \mathbf{w} \in \mathbb{R}^n$ are function parameters. Then the application of the unscented transformation

$$\left(\hat{\mathbf{x}}^*_{t-1},\, \mathbf{P}^*_{t-1}\right) \xrightarrow{f^*} \left(\hat{\mathbf{x}}^-_t,\, \mathbf{P}^-_t\right) \tag{2.62}$$

leads to:

$$\hat{\mathbf{x}}_t^- \approx \mathrm{E}[\, f(\mathbf{x}_{t-1})\,] \qquad (a\ priori \text{ state estimate}) \tag{2.63}$$

$$\mathbf{P}_t^- \approx \mathrm{cov}(\, f(\mathbf{x}_{t-1}) + \mathbf{w}_t\,) \qquad (a\ priori \text{ estimate error covariance}) \tag{2.64}$$

(ii) **Update step**

Similarly, define the augmented *a priori* state estimate $\hat{\mathbf{x}}_t^{-*} \in \mathbb{R}^{n+m}$ and error covariance $\mathbf{P}_t^{-*} \in \mathbb{R}^{(n+m)\times(n+m)}$ as:

$$\hat{\mathbf{x}}_t^{-*} = \begin{bmatrix} \hat{\mathbf{x}}_t^- \\ 0 \end{bmatrix} \qquad \mathbf{P}_t^{-*} = \begin{bmatrix} \mathbf{P}_t^- & 0 \\ 0 & \mathbf{R} \end{bmatrix} \tag{2.65}$$

and the augmented observation model $h^*\colon \mathbb{R}^{n+m} \to \mathbb{R}^m$ as:

$$h^*(\mathbf{x}, \mathbf{v}) = h(\mathbf{x}) + \mathbf{v} \tag{2.66}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{v} \in \mathbb{R}^m$ are function parameters. Then the application of the unscented transformation

$$\left(\, \hat{\mathbf{x}}_t^{-*},\, \mathbf{P}_t^{-*} \,\right) \xrightarrow{h^*} (\, \hat{\mathbf{z}}_t,\, \mathbf{Z}_t\,) \tag{2.67}$$

leads to:

$$\hat{\boldsymbol{z}}_t \approx \mathrm{E}[\, h(\mathbf{x}_t^-)\,] \qquad (\text{expected observation}) \tag{2.68}$$

$$\mathbf{Z}_t \approx \mathrm{cov}(\, h(\mathbf{x}_t^-) + \mathbf{v}_t\,) \qquad (\text{innovation covariance}) \tag{2.69}$$

$$\mathbf{Y}_t \approx \mathrm{cov}(\, \mathbf{x}_t^-,\, h(\mathbf{x}_t^-)\,) \qquad (\text{state and expected observation cross-covariance}) \tag{2.70}$$

where $\mathbf{x}_t^- = f(\mathbf{x}_{t-1}) + \mathbf{w}_t$. Note that the expression for $\mathbf{Y}_t$ is given by Equation (2.58) applied to the sigma points of this unscented transformation. The update equations of the filter are then as follows:

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t\,(\mathbf{z}_t - \hat{\mathbf{z}}_t) \qquad (a\ posteriori \text{ state estimate}) \tag{2.71}$$

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}\,\mathbf{Z}_t\,\mathbf{K}_t^{\mathrm{T}} \qquad (a\ posteriori \text{ estimate error covariance}) \tag{2.72}$$

$$\mathbf{K}_t = \mathbf{Y}_t\,\mathbf{Z}_t^{-1} \qquad (\text{Kalman gain factor}) \tag{2.73}$$

The remarks from Definition 2.4 on the properties of the initial state estimate $\hat{\mathbf{x}}_0$ and covariance $\mathbf{P}_0$ apply here too. Note that 0 in the definitions of the augmented states and covariances above denotes a zero matrix of an appropriate size.

From a theoretical perspective, similar to the EKF, the UKF is only guaranteed to provide optimal mean squared error estimates if the original Kalman filter assumptions are satisfied and

	Time	Space
Prediction step	$O(\, n^3 \,)$	$O(\, n^2 \,)$
Update step	$O(\, n^3 + n^2 m + n m^2 + m^3 \,)$	$O(\, n^2 + n m + m^2 \,)$

Table 2.3: Asymptotic computational complexity of the Unscented Kalman filter

the underlying process and observation models are linear. However, it has been demonstrated that the UKF provides reasonable estimates in many practical applications even with highly non-linear models with an accuracy that is consistently superior to the EKF, and furthermore, the UKF does not require the problematic computation of Jacobians. These properties led to widespread adoption of the UKF by the state estimation community over the past decade.

The main issue with the UKF is the computational complexity of the unscented transformations performed in both the prediction and the update step. The asymptotic computational complexity of a straightforward implementation of the UKF algorithm as in Definition 2.9 is summarised in Table 2.3. Although a more time-efficient variant of the UKF has been developed, which propagates the square root of the covariance matrix instead of the covariance matrix itself (Van Der Merwe and Wan, 2001), this variant replaces the $O(n^3)$ term in the asymptotic time complexity to $O(n^2)$ only for parameter estimation problems. For state estimation problems, the $O(n^3)$ term effectively prohibits application of the UKF to systems with a large number of state-space dimensions.

2.6 Ensemble Kalman Filter

The Ensemble Kalman filter (EnKF), originally proposed in (Evensen, 1994), is a Monte Carlo variant of the Kalman filter, which represents the state estimates and observations using a set of random samples, instead of the conventional means and covariances. Such a representation gives the EnKF three fundamental benefits compared to the other variants of the Kalman filter described in the previous sections (Evensen, 2009; Mandel, 2006). First, the computational time and space complexity of both the prediction and update operations scale linearly with the number of the state-space dimensions. Second, the EnKF approximates non-linear transformations of probability distributions with an accuracy only limited by the number of Monte Carlo samples, and therefore, potentially more precisely than both the EKF and UKF. Third, similarly to the UKF, the EnKF does not require Jacobians of the prediction and observation models, which are

generally difficult to compute, as discussed in Section 2.4.

The features of the EnKF led to its widespread adoption to estimation problems in the Earth sciences, where state spaces are often high-dimensional and models non-linear, and hence other variants of the Kalman filter are not applicable. For example, the EnKF became a de facto standard tool for data assimilation in numerical weather prediction (Evensen, 2009). On the other hand, the Monte Carlo representation of a probability distribution is always imprecise, even if the physical system is modelled precisely, which might be problematic in certain applications, such as life-critical systems. Additionally, although the EnKF has a better asymptotic time complexity than both the EKF and UKF with respect to the number of state-space dimensions, in practice the EnKF will be slower than the EKF and UKF in applications where the number of state-space dimensions is relatively small, which effectively prohibits its use in embedded or energy-constrained systems.

In the EnKF, the state estimate and the associated error at a time t is represented using an ordered set of random samples $\mathbf{x}_t^1, \ldots, \mathbf{x}_t^N \in \mathbb{R}^n$, called an *ensemble*, which are organised in a matrix $\mathbf{X}_t \in \mathbb{R}^{n \times N}$ as:

$$\mathbf{X}_t \;=\; \big[\, \mathbf{x}_t^1, \ldots, \mathbf{x}_t^N \,\big] \tag{2.74}$$

where n is the number of state-space dimensions and $N > 1$ is the number of random samples (ensemble members). Note that the ensemble $\mathbf{X}_t$ effectively approximates the probability distribution of the possible value of a true system state $\mathbf{x}_t$; the approximation of the distribution is perfect in the limit on an infinite ensemble. In this sense, the mean of the ensemble models the mean state estimate, and the spread of ensemble members models the error statistics associated with the estimate, which was previously characterised by the state estimate error covariance. The ensemble $\mathbf{X}_t$ can be easily transformed using any function $g \colon \mathbb{R}^n \to \mathbb{R}^m$, to obtain a *transformed ensemble* $g(\mathbf{X}_t) \in \mathbb{R}^{m \times N}$ as:

$$g(\mathbf{X}_t) \;=\; \big[\, g(\mathbf{x}_t^1), \ldots, g(\mathbf{x}_t^N) \,\big] \tag{2.75}$$

Such an ensemble $g(\mathbf{X}_t)$ approximates the probability distribution of the random vector $g(\mathbf{x}_t)$.

Similarly to the state estimate, an observation of the physical system and the associated error at time t is represented using an ordered set of random samples $\mathbf{z}_t^1, \ldots, \mathbf{z}_t^N \in \mathbb{R}^m$ organised in a matrix $\mathbf{Z}_t \in \mathbb{R}^{m \times N}$:

$$\mathbf{Z}_t \;=\; \big[\, \mathbf{z}_t^1, \ldots, \mathbf{z}_t^N \,\big] \tag{2.76}$$

where m denotes the number of observation dimensions. Such an *observation ensemble* $\mathbf{Z}_t$ can be computed from a base observation vector $\mathbf{z}_t \in \mathbb{R}^m$ by adding randomly generated vectors

with statistical properties of the observation noise from Definition 2.1 of the linear discrete-time dynamical system model, i.e. a random sampling scheme which can be symbolically denoted as:

$$\mathbf{z}_t^i \leftarrow \mathcal{N}(\mathbf{z}_t, \mathbf{R}) \tag{2.77}$$

for every $i = 1, \ldots, N$. The observation ensemble $\mathbf{Z}_t$ then has the following property:

$$\mathrm{cov}(\mathbf{Z}_t) \simeq \mathbf{R} \tag{2.78}$$

Various methods have been proposed in the literature to improve the quality of the outcome of the sampling procedure, in particular so that its statistical properties better reflect the desired probability distribution parameters. For example, it is possible to shift each of the observation ensemble members to ensure that their mean is equivalent to the observation $\mathbf{z}_t$. Formally, such a refined observation ensemble $\mathbf{Z}_t^* \in \mathbb{R}^{m \times N}$ can be computed as follows:

$$\mathbf{Z}_t^* = \left[\mathbf{z}_t^1 - \mathrm{E}[\mathbf{Z}_t] + \mathbf{z}_t, \ldots, \mathbf{z}_t^N - \mathrm{E}[\mathbf{Z}_t] + \mathbf{z}_t \right] \tag{2.79}$$

This and other similar methods are analysed in detail in (Evensen, 2009).

Note that $\mathrm{cov}(\mathbf{A}, \mathbf{B})$ in this context denotes a sample covariance between two ensembles $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]$ and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_N]$, computed as follows:

$$\mathrm{cov}(\mathbf{A}, \mathbf{B}) = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{a}_i - \mathrm{E}[\mathbf{A}])(\mathbf{b}_i - \mathrm{E}[\mathbf{B}])^{\mathrm{T}} \tag{2.80}$$

and $\mathrm{cov}(\mathbf{A})$ is just a shorthand for $\mathrm{cov}(\mathbf{A}, \mathbf{A})$. In fact, the sample covariance represents an estimate of the true covariance $\mathrm{cov}(\mathbf{a}, \mathbf{b})$, as defined in Equation (2.7), between the two random variables $\mathbf{a}$ and $\mathbf{b}$ whose estimates are represented by the respective ensembles, formally:

$$\mathrm{cov}(\mathbf{a}, \mathbf{b}) \simeq \mathrm{cov}(\mathbf{A}, \mathbf{B}) \tag{2.81}$$

Similarly, $\mathrm{E}[\mathbf{A}]$ denotes a simple arithmetic mean of an ensemble $\mathbf{A}$ computed as:

$$\mathrm{E}[\mathbf{A}] = \frac{1}{N} \sum_{i=1}^{N} \mathbf{a}_i \tag{2.82}$$

which represents an estimate of the true expected value $\mathrm{E}[\mathbf{a}]$ of the random variable $\mathbf{a}$, i.e.

$$\mathrm{E}[\mathbf{A}] \simeq \mathrm{E}[\mathbf{a}] \tag{2.83}$$

And finally, $\mathbf{C} \simeq \mathbf{D}$ denotes an asymptotic equality of terms $\mathbf{C}$ and $\mathbf{D}$ in the limit of an infinite ensemble, formally $\lim_{N \to +\infty}(\mathbf{C} - \mathbf{D}) = 0$.

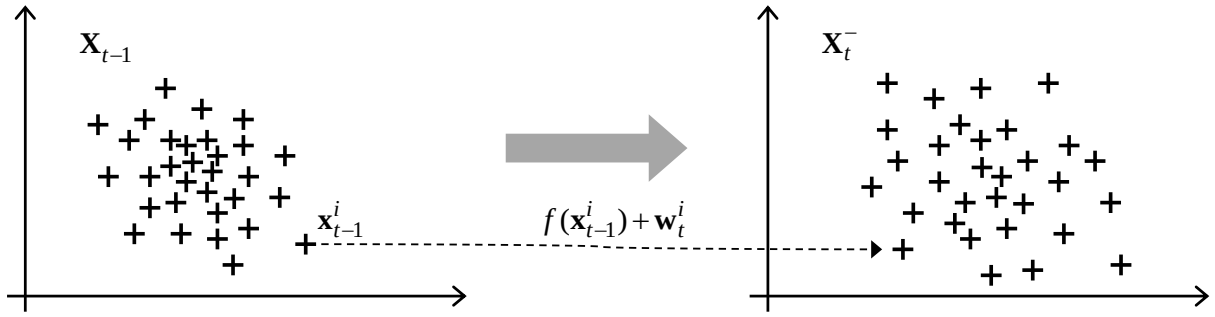


Fig. 2.5: Prediction step of the Ensemble Kalman filter.

Definition 2.10 (Ensemble Kalman filter). Suppose a physical system is modelled as a non-linear discrete-time dynamical system from Definition 2.6, with independent zero-mean process and observation noise as in Definition 2.3. The estimate $\mathbf{X}_t \in \mathbb{R}^{n \times N}$ of the system state at a time $t \in \mathbb{N}$, given all the observations $\mathbf{z}_1, \ldots, \mathbf{z}_t$ up to time t, can be computed recursively from the previous state estimate ensemble $\mathbf{X}_{t-1} \in \mathbb{R}^{n \times N}$ and the current observation ensemble $\mathbf{Z}_t \in \mathbb{R}^{m \times N}$ using the following equations:

(i) Prediction step

$$\mathbf{X}_t^- = f(\mathbf{X}_{t-1}) + \mathbf{W}_t \qquad (a\ priori \text{ state estimate}) \qquad (2.84)$$

$$\mathbf{W}_t = [\mathbf{w}_t^1, \ldots, \mathbf{w}_t^N] \qquad (\text{process noise compensation}) \qquad (2.85)$$

where $\mathbf{w}_t^i \in \mathbb{R}^n$ are randomly and independently generated vectors $\mathbf{w}_t^i \leftarrow \mathcal{N}(0, \mathbf{Q})$ for every $i = 1, \ldots, N$, that compensate for the process noise $\mathbf{w}_t$.

(ii) Update step

$$\mathbf{X}_t = \mathbf{X}_t^- + \mathbf{K}_t(\mathbf{Z}_t - h(\mathbf{X}_t^-)) \qquad (a\ posteriori \text{ state estimate}) \qquad (2.86)$$

$$\mathbf{K}_t = \operatorname{cov}(\mathbf{X}_t^-, h(\mathbf{X}_t^-))\,[\operatorname{cov}(h(\mathbf{X}_t^-)) + \mathbf{R}]^{-1} \qquad (\text{Kalman gain factor}) \qquad (2.87)$$

As similarly noted in Definition 2.4 of the Kalman filter, the initial ensemble $\mathbf{X}_0$ is assumed to be sampled randomly from the Gaussian probability distribution of the initial state described in Definition 2.1.

The EnKF, in particular its update step, is based on the same assumptions as the traditional Kalman filter, such as that all the involved probability distributions are Gaussian, the observation model $h\colon \mathbb{R}^n \to \mathbb{R}^m$ is linear, and that the *a priori* state estimate $\mathbf{X}_t^-$ and the observation $\mathbf{Z}_t$ are statistically independent. Additionally, due to its Monte Carlo nature, the EnKF is only guaranteed to provide optimal mean squared error estimates in the limit of an infinite ensemble.

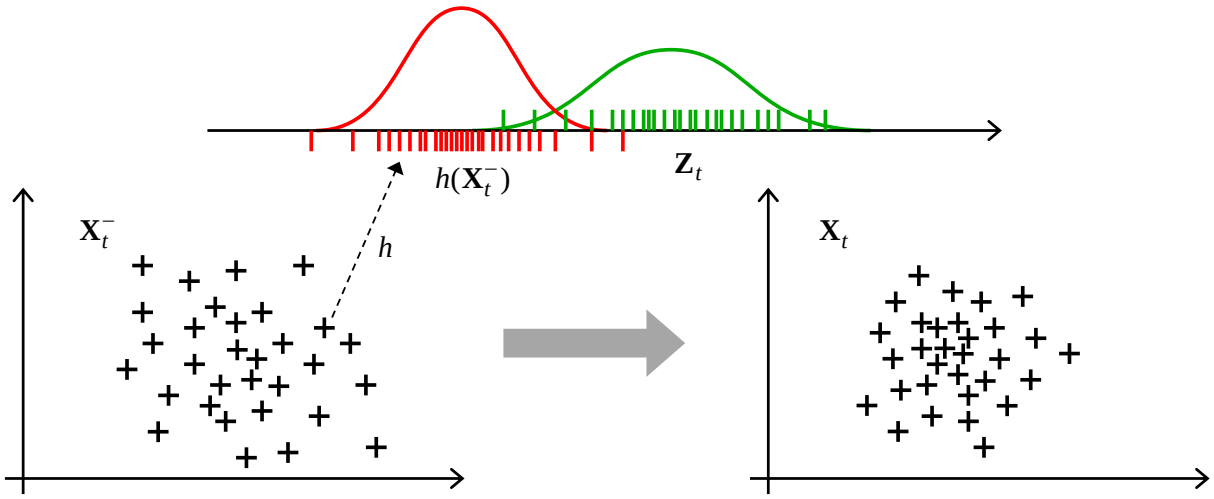


Fig. 2.6: Update step of the Ensemble Kalman filter.

	Time	Space
Prediction step	$O(\, n^2N + nN \,)$	$O(\, n^2 + nN \,)$
Update step	$O(\, nmN + nm^2 + m^2N + m^3 \,)$	$O(\, nN + nm + mN + m^2 \,)$

Table 2.4: Asymptotic computational complexity of the Ensemble Kalman filter

Although these assumptions might seem quite restrictive, like other Kalman filter variants, the EnKF with a moderate number of ensemble members provides reasonable state estimates even in situations where the assumptions are not exactly satisfied, that is, in most practical applications. Illustrations of the prediction step and the update step of the EnKF in a dynamical system with $n = 2$ and $m = 1$ are provided in Figures 2.5 and 2.6, respectively.

The asymptotic computational complexity of a straightforward implementation of the EnKF algorithm from Definition 2.10 is summarised in Table 2.4. Although the time and space complexity of the prediction step involves $O(n^2N)$ and $O(n^2)$ terms, respectively, these terms only describe the complexity of generation of the process noise compensation samples $\mathbf{W}_t$ in case of a general process noise covariance matrix $\mathbf{Q}$. In most practical applications, $\mathbf{Q}$ is a diagonal matrix, and therefore both these quadratic complexity terms will not be present. In any case, the covariance matrix $\mathbf{Q}$ here is assumed constant over time, and therefore, its Cholesky decomposition necessary for the generation of random samples $\mathbf{W}_t$ can be precomputed. Also, it is assumed that the functions f and h can be evaluated in $O(n)$ and $O(n + m)$ time and space, respectively. Note that various algebraic formulations of the update step with various

asymptotic complexities have been proposed in the literature, which might be more efficient for particular applications (Evensen, 2003; Mandel, 2006); here the order of matrix multiplications is optimised for $n \gg m$ and $n \gg N$.

Aside from the computational advantages of the EnKF described in the beginning of this section, the ensemble representation of the state estimates and observations brings another fundamental advantage - if the ensemble members have a fixed order, one can estimate the cross-covariance between two seemingly unrelated ensembles, simply by applying Equation (2.80). This is the principal insight provided in this thesis, leading to the development of a new CPI-EnKF and Augmented EnKF filters that support correlations between errors associated with the state estimates and observations. Additionally, the literature mentions virtually no practical applications of the EnKF in many areas of engineering, such as in robotics, most likely due to the availability of algorithms that are faster in lower-dimensional problems, such as the UKF. This thesis aims to challenge this status quo, by demonstrating that the EnKF offers some unprecedented features that fully justify its use in engineering, while arguing that the performance penalty in lower-dimensional problems is practically negligible on modern hardware.

2.7 Particle Filter

The particle filter, also known as the sequential Monte Carlo method, is a popular recursive non-linear Bayesian inference algorithm (Doucet and Johansen, 2009). Although traces of the method date back to 1950s (Hammersley and Morton, 1954), the current formulation of the filter was first introduced in a seminal paper by (Gordon et al., 1993). Similarly to the EnKF, the particle filter represents the current state estimate and its uncertainty using a set of random samples drawn from the state space, which is referred to as *particles*. As such, the particle filter is an approximate Monte Carlo algorithm that provides optimal solutions only in the limit of an infinite number of particles. The particle filter is reviewed in this chapter purely to illustrate its differences from the EnKF.

Similarly to the Kalman filter, the operation of the particle filter can also be divided into two logical steps: the prediction step and the update step. In the prediction step, the particles are propagated to the next time step to model a so-called *proposal distribution* (effectively an *a priori* state estimate), very much the same way the EnKF performs the prediction step. On the other hand, the update step of the particle filter is very different compared to the EnKF. Each particle representing the proposal distribution is assigned a weight that depends

on the probability distribution of the observation and then the particles are *resampled* based on these weights in order to obtain the *target distribution* (an *a posteriori* state estimate), using an algorithm such as the sequential importance resampling (SIR). Formally, the proposal distribution at a time t is represented using a set of N particles:

$$\left\{ \bar{\mathbf{x}}_t^{(i)} \mid i = 1, \ldots, N \right\} \tag{2.88}$$

In the update step, each particle $\bar{\mathbf{x}}_t^{(i)} \in \mathbb{R}^n$ is assigned a weight coefficient $\bar{\mathbf{w}}_t^{(i)} \in \mathbb{R}$ that corresponds to the probability of the observation $\mathbf{z}_t \in \mathbb{R}^m$ given the true state $\bar{\mathbf{x}}_t^{(i)}$, i.e.

$$\bar{\mathbf{w}}_t^{(i)} = \frac{1}{N} P\left(\mathbf{z}_t \,|\, \bar{\mathbf{x}}_t^{(i)} \right) \tag{2.89}$$

The weights are then normalised using the following rule:

$$\mathbf{w}_t^{(i)} = \frac{\bar{\mathbf{w}}_t^{(i)}}{\sum_{i=1}^{N} \bar{\mathbf{w}}_t^{(i)}} \tag{2.90}$$

so that $\sum_{i=1}^{N} \mathbf{w}_t^{(i)} = 1$. Finally, N particles are randomly drawn from the set $\left\{ \bar{\mathbf{x}}_t^{(i)} \mid i = 1, \ldots, N \right\}$ with probabilities proportional to the corresponding weights $\left\{ \mathbf{w}_t^{(i)} \mid i = 1, \ldots, N \right\}$, leading to the final set of particles:

$$\left\{ \mathbf{x}_t^{(i)} \mid i = 1, \ldots, N \right\} \tag{2.91}$$

that represents the target probability distribution.

Note that in the update step of the particle filter, no particles are being altered, only particles with higher weights are kept or even duplicated while particles with lower weights are discarded. Particle filters can operate with arbitrary (non-Gaussian) probability distributions and non-linear dynamical system models with an arbitrary precision, given a sufficient number of particles. Unfortunately, the number of necessary particles grows with the complexity of the probability distributions involved (e.g. number of dimensions, non-zero higher moments or multimodality), which is also known as the *particle depletion* or *degeneracy* problem.

The update step of the particle filter is in a direct contrast with the update step of the EnKF, where each of the ensemble members is linearly transformed using the corresponding observation ensemble member and the Kalman gain factor, instead of resampling. Regardless of the *a priori* state estimate and observation probability distributions, the distribution of the resulting *a posteriori* state estimate tends to be Gaussian due to the fuzzy central limit theorem, and consequently, it can be represented using a constant number of ensemble members without

degeneracy. Additionally, resampling would make it impossible to maintain a pairwise cross-correlation between two ensembles, which is the core feature of the EnKF around which the contribution of this thesis is built. This is explained in detail in Chapter 5.

2.8 Chapter Summary

This chapter formally defined the Kalman filter and its arguably four most important variants: the Information filter, the Extended Kalman filter (EKF), the Unscented Kalman filter (UKF) and the Ensemble Kalman filer (EnKF), and discussed in detail the most important features of the filters relevant to this thesis - their ability to operate with non-linear models and their computational complexity. The chapter also introduced the formalism and notation that will be used throughout the rest of this thesis, and reviewed the particle filter, in order to illustrate its differences from the EnKF.

Chapter 3

Correlated Noises

All the variants of the Kalman filter described in Chapter 2 are based on the same formal model, which assumes that both the process and observation noises are white and independent. This implies that both the noises are sequentially uncorrelated and uncorrelated to each other and that state estimate errors are not correlated to observation errors during the update step. Unfortunately, in many practical applications, the physical system does not fit such a model, which can lead to poor performance of the conventional Kalman filter, in particular that it might provide overconfident state estimates. If such estimates were to affect the decisions of a safety-critical system, for example, autonomous control of a vehicle, the consequences could be severe.

This chapter reviews the common types of correlation in or between the state and observation noises, provides practical examples where they appear, and describes the state-of-the-art approaches that enable consistent state estimation in their presence. Although the approaches are described in the context of the conventional Kalman filter, most of them can be applied equally with all the variants of the Kalman filter described in Chapter 2; exceptions to this rule are pointed out explicitly.

3.1 Correlated Process and Observation Noise

The conventional Kalman filter is based on a model of the underlying dynamical system that assumes that both the process noise and the observation noise are mutually independent, and therefore, mutually uncorrelated - see Definition 2.3-(ii). This section will relax this assumption, and describe generalised Kalman filter equations that provide optimal and consistent state

estimates even if the process noise and the observation noise are correlated. For example, such a generalisation is useful in the following scenario (Simon, 2006, p. 184): Assume one has to estimate a position of an aircraft by modelling its flight dynamics and measuring the wind speed using an anometer. An occasional wind gust will affect both the error in the aircraft dynamics model (process noise) and the error in the measurements of the anometer (observation noise) in a similar, correlated, way. As such, the traditional Kalman filter might provide inconsistent state estimates but the generalised Kalman filter equations described in this section will provide consistent estimates. Moreover, these equations serve as a building block for several other methods to accommodate different types of correlation in the noises, such as sequential correlation in the observation noise described in Section 3.3.2.

The literature distinguishes two main types of a correlation between the process noise and the observation noise, depending on at which time steps are the noise terms correlated. Recall Definition 2.1 of the linear discrete-time dynamical system model:

$$\mathbf{x}_t = \mathbf{F}\,\mathbf{x}_{t-1} + \mathbf{w}_t \tag{3.1}$$

$$\mathbf{z}_t = \mathbf{H}\,\mathbf{x}_t + \mathbf{v}_t \tag{3.2}$$

where the terms $\mathbf{w}_t$ and $\mathbf{v}_t$ represent the process and the observation noise, respectively. In the type I of correlation, the process noise term $\mathbf{w}_t$ is assumed correlated with the subsequent observation noise term $\mathbf{v}_t$, which requires a redefinition of the update step. In the type II, the process noise term $\mathbf{w}_t$ is assumed correlated with the previous observation noise term $\mathbf{v}_{t-1}$, which requires a redefinition of the prediction step.

Definition 3.1 (Kalman filter for correlated process and observation noise - type I). Suppose a physical system is modelled as a linear discrete-time dynamical system from Definition 2.1, with a process and observation noise as before in Definition 2.3, except that the assumption (ii) is replaced by the following assumption: the process noise sequence $\{\mathbf{w}_t\}$ and the observation noise sequence $\{\mathbf{v}_t\}$ are mutually correlated as:

$$\operatorname{cov}(\mathbf{w}_i, \mathbf{v}_j) = \begin{cases} \mathbf{C} & i = j \\ 0 & i \neq j \end{cases} \tag{3.3}$$

where $\mathbf{C} \in \mathbb{R}^{n \times m}$. The update step of the Kalman filter from Definition 2.4 then needs to be

replaced with the following equations:

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t \left(\mathbf{z}_t - \mathbf{H}\, \hat{\mathbf{x}}_t^- \right) \tag{3.4}$$

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t \left(\mathbf{H}\, \mathbf{P}_t^- + \mathbf{C}^{\mathrm{T}} \right) \tag{3.5}$$

$$\mathbf{K}_t = \left(\mathbf{P}_t^-\, \mathbf{H}^{\mathrm{T}} + \mathbf{C} \right) \left[\mathbf{H}\, \mathbf{P}_t^-\, \mathbf{H}^{\mathrm{T}} + \mathbf{H}\mathbf{C} + \mathbf{C}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}} + \mathbf{R} \right]^{-1} \tag{3.6}$$

The prediction step equations of the filter, as well as all the other remarks and the notation of variables, are as before in Definition 2.4. The Kalman filter with this set of update equations is also referred to as the *Generalised Kalman filter*.

Definition 3.2 (Kalman filter for correlated process and observation noise - type II). Suppose a physical system is modelled as a linear discrete-time dynamical system from Definition 2.1, with a process and observation noise as before in Definition 2.3, except that the assumption (ii) is replaced by the following assumption: the process noise sequence $\{\mathbf{w}_t\}$ and the observation noise sequence $\{\mathbf{v}_t\}$ are mutually correlated as:

$$\mathrm{cov}(\mathbf{w}_i, \mathbf{v}_{j-1}) = \begin{cases} \mathbf{C} & i = j \\ 0 & i \neq j \end{cases} \tag{3.7}$$

where $\mathbf{C} \in \mathbb{R}^{n \times m}$. The prediction step of the Kalman filter from Definition 2.4 then needs to be replaced with the following equations:

$$\hat{\mathbf{x}}_t^- = \mathbf{F}\, \hat{\mathbf{x}}_{t-1} + \mathbf{C} \left[\mathbf{H}\mathbf{P}_{t-1}^- \mathbf{H}^{\mathrm{T}} + \mathbf{R} \right]^{-1} \left(\mathbf{z}_{t-1} - \mathbf{H}\, \hat{\mathbf{x}}_{t-1}^- \right) \tag{3.8}$$

$$\mathbf{P}_t^- = \mathbf{F}\, \mathbf{P}_{t-1}\, \mathbf{F}^{\mathrm{T}} + \mathbf{Q} - \mathbf{C} \left[\mathbf{H}\mathbf{P}_{t-1}^- \mathbf{H}^{\mathrm{T}} + \mathbf{R} \right]^{-1} \mathbf{C}^{\mathrm{T}} - \mathbf{F}\mathbf{K}_{t-1}\mathbf{C}^{\mathrm{T}} - \mathbf{C}\mathbf{K}_{t-1}^{\mathrm{T}}\mathbf{F}^{\mathrm{T}} \tag{3.9}$$

The update step equations of the filter, as well as all the other remarks and the notation of variables, are as before in Definition 2.4.

For brevity, both Definitions 3.1 and 3.2 assume that the correlation term $\mathbf{C}$ is time-invariant, but in practice, it can change over time, similarly to the noise covariance terms $\mathbf{Q}$ and $\mathbf{R}$. Note that both the filters are in fact generalisations of the Kalman filter - if the correlation term $\mathbf{C}$ is zero, they both reduce to the traditional Kalman filter described in Definition 2.4. Both these generalisations were originally derived in (Kalman, 1963) and they are both guaranteed to provide optimal and consistent state estimates if all the assumptions are satisfied. A detailed derivation of the equations in Definition 3.1 can also be found in (Simon, 2006, p. 184), as well as in (Brown and Hwang, 2012; Nieto and Guerrero, 1995). Variants of all the equations

based on the more general linear dynamical system model as in Equation (2.1) can be found in (Maybeck, 1979)[p. 246].

Although both the generalisations from Definitions 3.1 and 3.2 are described in the context of the basic Kalman filter, they can also be easily applied in the context of the Extended Kalman filter (EKF), where the process model F and the observation model H need to replaced with the Jacobians $\mathcal{F}_t$ and $\mathcal{H}_t$, respectively, similarly to Definition 2.7 of the EKF. With respect to the Information filter and the Unscented Kalman filter (UKF), corresponding variants of these filters that support equivalent forms of correlation between the process and observation noise could theoretically be developed, but the author of this thesis is not aware of any literature describing such variants, and their derivation is beyond the scope of the thesis. However, both the generalisations of the Kalman filter cannot be directly applied in the context of the Ensemble Kalman filter (EnKF). In the case of the type I correlation, it will be demonstrated in Section 6.3.2 that the update rule provided in Definition 3.1 cannot be applied in the EnKF, i.e. such a Generalised EnKF is not a valid filter. On the other hand, the prediction rule for the type II correlation provided in Definition 3.2 is not necessary at all in the EnKF, as it can be replaced by a method described in Section 5.1. Note that both these findings constitute contributions provided by this thesis.

3.2 Sequentially Correlated Process Noise

In theory, the Kalman filter assumes that the model of the underlying dynamical system is perfect. In practice, a designer of virtually every application of the Kalman filter needs to make a choice of the level of detail with which a dynamical system is modelled, primarily to choose the state variables and the time step length, while the effects of any unmodelled dynamics are considered a part of the process noise. Typically, this choice is done in such a way that the effects of unmodelled dynamics appear sequentially uncorrelated at the process time steps, and hence the process noise can be modelled as white. However, in certain situations, this approach is rather impractical as it requires development of complex process models, while it is easier and sufficiently accurate to just assume the process noise is sequentially correlated. For example, when estimating a position and attitude of an aircraft, it is probably easier to model random wind gusts that affect the motion of the aircraft as a sequentially correlated process noise, rather than to model the wind parameters as part of the state space and predict their effects on the aircraft motion (Crassidis and Junkins, 2011, p. 226). A sequentially correlated noise is often

referred to in the literature as *coloured noise*, as well as *correlated noise*, *time-correlated noise*, *auto-correlated noise*, or *noise with serial correlation*. This section will describe variants of the Kalman filter that provide optimal state estimates in the presence of such a coloured process noise.

For simplicity, this section only considers the most elementary form of a sequential correlation in the noise sequence, modelled as a wide-sense-stationary zero-mean first-order autoregressive process (an AR(1) process), and the term *coloured noise* is used for this type of a random process only.

Definition 3.3 (Coloured noise process). An n-dimensional random process $\{\mathbf{w}_t\}$ is called a *coloured noise* if and only if:

(i) For every $t \in \mathbb{N}$, the random vector $\mathbf{w}_t \in \mathbb{R}^n$ has zero mean and a finite covariance $\mathbf{Q} \in \mathbb{R}^{n\times n}$, i.e.

$$\mathrm{E}[\mathbf{w}_t] = 0 \tag{3.10}$$

$$\mathrm{cov}(\mathbf{w}_t) = \mathbf{Q} \tag{3.11}$$

(ii) For every $t \in \mathbb{N}$:

$$\mathbf{w}_{t+1} = \boldsymbol{\Psi}\,\mathbf{w}_t + \varepsilon_t \tag{3.12}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{n\times n}$ is an auto-correlation coefficient matrix, $\mathbf{w}_t$ and $\varepsilon_t \in \mathbb{R}^n$ are mutually independent random vectors, and the sequence $\{\varepsilon_t\}$ is a white noise process with zero mean and a constant finite covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$, i.e.

$$\mathrm{cov}(\,\mathbf{w}_t,\, \varepsilon_t\,) = 0 \tag{3.13}$$

$$\mathrm{cov}(\,\varepsilon_t\,) = \boldsymbol{\Sigma} \tag{3.14}$$

Note that if $\boldsymbol{\Psi} = 0$, then the coloured noise process $\{\mathbf{w}_t\}$ reduces to a white noise process. Also, be aware that the terms *process noise* and *noise process*, although similar, have two different meanings. Although both the auto-correlation coefficient matrix $\boldsymbol{\Psi}$ and the covariance matrix $\boldsymbol{\Sigma}$ are assumed constant over time, all the methods described in this and the following sections can be directly extended to support time-varying matrices $\boldsymbol{\Psi}_t$ and $\boldsymbol{\Sigma}_t$ instead, similarly to other parameter matrices ($\mathbf{F}$, $\mathbf{H}$, $\mathbf{Q}$, $\mathbf{R}$, etc.) used throughout this thesis, as noted in Section 2.1.

In order to accommodate a coloured process noise in the Kalman filter, the filter equations do not need to be redefined, it only suffices to augment the dynamical system model. This

procedure was introduced in (Bucy and Joseph, 1968) and is also described in (Simon, 2006, p. 188). The equations of the linear discrete-time dynamical system model from Definition 2.1:

$$\mathbf{x}_t = \mathbf{F}\,\mathbf{x}_{t-1} + \mathbf{w}_t \tag{3.15}$$

$$\mathbf{z}_t = \mathbf{H}\,\mathbf{x}_t + \mathbf{v}_t \tag{3.16}$$

and the equation characterising an n-dimensional coloured process noise sequence $\{\mathbf{w}_t\}$:

$$\mathbf{w}_{t+1} = \mathbf{\Psi}\,\mathbf{w}_t + \varepsilon_t \tag{3.17}$$

can be combined into a single system of equations:

$$\begin{bmatrix} \mathbf{x}_t \\ \mathbf{w}_{t+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F} & I \\ 0 & \mathbf{\Psi} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{t-1} \\ \mathbf{w}_t \end{bmatrix} + \begin{bmatrix} 0 \\ \varepsilon_t \end{bmatrix} \tag{3.18}$$

$$\mathbf{z}_t = \begin{bmatrix} \mathbf{H} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_t \\ \mathbf{w}_{t+1} \end{bmatrix} + \mathbf{v}_t \tag{3.19}$$

which corresponds to a system of equations describing an augmented dynamical system model:

$$\mathbf{x}_t^* = \mathbf{F}^*\,\mathbf{x}_{t-1}^* + \mathbf{w}_t^* \tag{3.20}$$

$$\mathbf{z}_t = \mathbf{H}^*\,\mathbf{x}_t^* + \mathbf{v}_t \tag{3.21}$$

where $\mathbf{x}_t^* \in \mathbb{R}^{2n}$ represents the augmented state vector, $\mathbf{F}^* \in \mathbb{R}^{2n\times 2n}$ the augmented process model, $\mathbf{H}^* \in \mathbb{R}^{m\times 2n}$ the augmented observation model, and $\mathbf{w}_t^* \in \mathbb{R}^{2n}$ represents the augmented process noise term with a covariance $\mathbf{Q}^* = \mathrm{cov}(\mathbf{w}_t^*) \in \mathbb{R}^{2n\times 2n}$ defined as:

$$\mathbf{Q}^* = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{\Sigma} \end{bmatrix} \tag{3.22}$$

where $\mathbf{\Sigma} = \mathrm{cov}(\varepsilon_t) \in \mathbb{R}^{n\times n}$ for all $t \in \mathbb{N}$. Note that I and 0 above denote the identity and zero matrices of an appropriate dimension, respectively. The augmented process noise sequence $\{\mathbf{w}_t^*\}$ is indeed a white noise, the augmented system meets all the necessary assumptions, and therefore, the Kalman filter can estimate its state. However, this comes at a price of an increased computational complexity, as the number of state-space dimensions of the problem has doubled.

The state of such an augmented system can be estimated using any variant of the Kalman filter described in Section 2.2. In the context of the EKF, the UKF and the EnKF, the non-linear functions $f^*\colon \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ and $h^*\colon \mathbb{R}^{2n} \to \mathbb{R}^m$ corresponding to the augmented process model $\mathbf{F}^*$

and the augmented observation model $\mathbf{H}^*$, respectively, can be defined as:

$$f^*(\mathbf{x}, \mathbf{w}) = \begin{bmatrix} f(\mathbf{x}) + \mathbf{w} \\ \boldsymbol{\Psi}\mathbf{w} \end{bmatrix} \tag{3.23}$$

$$h^*(\mathbf{x}, \mathbf{w}) = h(\mathbf{x}) \tag{3.24}$$

where $\mathbf{x}, \mathbf{w} \in \mathbb{R}^n$ are function parameters, and $f\colon \mathbb{R}^n \to \mathbb{R}^n$ and $h\colon \mathbb{R}^{2n} \to \mathbb{R}^m$ are the original process and observation model functions, respectively, as in Definition 2.6 of the non-linear discrete-time dynamical system model. This augmentation procedure of the process model function is not to be confused with a similar augmentation performed by the UKF (see Definition 2.9); both the augmentations need to be performed independently.

The concept of augmentation of the system model in order to accommodate for a coloured process noise can be easily extended to the more general time-evolution model of the dynamic system from Equation (2.1), as shown in (Crassidis and Junkins, 2011, p. 224), and also to a higher-order sequential correlation in the process noise. Furthermore, alternative methods have been proposed that do not increase the number of dimensions of the system model, while still supporting such a higher-order sequential correlation in the process noise (Jiang et al., 2010).

Note that in the context of the EnKF it is possible to compute consistent state estimates in the presence of sequentially correlated process noise even without state augmentation, using a method described in Section 5.1. This is especially useful in applications with a high number of state-space dimensions n, where the performance penalty associated with the state augmentation procedure would have a significant impact. This new method represents one of the contributions provided by this thesis.

3.3 Sequentially Correlated Observation Noise

One of the important assumptions of the Kalman filter and its variants described in Chapter 2 is the assumption that the observation noise is white, and therefore, sequentially uncorrelated. However, in reality, measurements of physical phenomena performed in succession often exhibit similar errors, in particular if the period between the measurements is short compared to a period with which the underlying sources of the error change. (Bryson and Johansen, 1965) characterised this problem as:

> *"Many practical systems exist in which the correlation times of the random measurement errors are* ***not*** *short compared to times of interest in the system; for brevity*

such errors are called 'colored' noise."

This section will describe variants of the Kalman filter that provide optimal state estimates in the presence of such measurement errors.

The problem of a sequential correlation in the observation noise occurs in many practical applications, for example, in radar-based target tracking an effect known as the *target scintillation* (or *glint*) causes the measurement errors to have a finite bandwidth (Skolnik, 2008; Wu and Chang, 1996), in navigation using the global positioning system (GPS) because the sources of the GPS error change relatively slowly over time (Petovello et al., 2005; Kaplan and Hegarty, 2006; Geier et al., 1995), in spacecraft position/attitude estimation using sensors affected by vibrations (Kumar and Crassidis, 2007) or even in specialised applications such as the analysis of records of ion channel operation in cell membranes (De Gunst et al., 2001). On the other hand, if the period between measurements is sufficiently long, the measurement error typically has enough time to *decorrelate* and hence it can be modelled as a white noise. Due to this premise, the majority of practical applications of the Kalman filter can safely ignore the sequential correlation in the observation noise.

From a technical standpoint, the presence of a sequential correlation in the observation noise will express itself as a correlation between the error in the *a priori* state estimate and the observation error in the update step of the Kalman filter. Therefore, one could theoretically apply the Generalised Kalman filter equations from Definition 3.1 to perform the update step in a consistent and optimal way. Unfortunately, there is no simple way to compute the necessary cross-covariance between the *a priori* state estimate error and the observation error, and therefore, more involved variants of the Kalman filter are required.

The literature recognises two main approaches to accommodate a sequentially correlated observation noise in the Kalman filter, *state augmentation* and *measurement differencing*, both of whom will be described in this section. As in Section 3.2, only the most elementary first-order sequential correlation in the noise process from Definition 3.3 will be considered, and such a process will be called again *coloured noise*. The parameters of such a first-order model can be estimated from the measurements using a method described in (Wu and Chang, 1996). If the first-order model cannot approximate the real measurement noise well enough, it is also possible to split the noise into sub-bands, model each of them separately using a first-order model, and apply any of the filtering approaches for the first-order correlated noise, as shown in (Chang and Wu, 2001).

3.3.1 State Augmentation

The state augmentation procedure to accommodate for a coloured observation noise in the Kalman filter was originally suggested in (Kalman, 1960) in an example, and further developed in (Kalman, 1963). The procedure is principally similar to the approach described in the case of a coloured process noise in Section 3.1. The system model equations from Definition 2.1:

$$\mathbf{x}_t \;=\; \mathbf{F}\,\mathbf{x}_{t-1} + \mathbf{w}_t \tag{3.25}$$

$$\mathbf{z}_t \;=\; \mathbf{H}\,\mathbf{x}_t + \mathbf{v}_t \tag{3.26}$$

and the equation characterising the coloured observation noise sequence $\{\mathbf{v}_t\}$ from Definition 3.3:

$$\mathbf{v}_t \;=\; \mathbf{\Psi}\,\mathbf{v}_{t-1} \;+\; \varepsilon_{t-1} \tag{3.27}$$

can be combined into a single system of equations:

$$\begin{bmatrix} \mathbf{x}_t \\ \mathbf{v}_t \end{bmatrix} = \begin{bmatrix} \mathbf{F} & 0 \\ 0 & \mathbf{\Psi} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{t-1} \\ \mathbf{v}_{t-1} \end{bmatrix} + \begin{bmatrix} \mathbf{w}_t \\ \varepsilon_{t-1} \end{bmatrix} \tag{3.28}$$

$$\mathbf{z}_t \;=\; \begin{bmatrix} \mathbf{H} & I \end{bmatrix} \begin{bmatrix} \mathbf{x}_t \\ \mathbf{v}_t \end{bmatrix} + 0 \tag{3.29}$$

which corresponds to an augmented system model:

$$\mathbf{x}^*_t \;=\; \mathbf{F}^*\,\mathbf{x}^*_{t-1} + \mathbf{w}^*_t \tag{3.30}$$

$$\mathbf{z}_t \;=\; \mathbf{H}^*\,\mathbf{x}^*_t + \mathbf{v}^*_t \tag{3.31}$$

where $\mathbf{x}^*_t \in \mathbb{R}^{n+m}$ represents the augmented state, $\mathbf{F}^* \in \mathbb{R}^{(n+m)\times(n+m)}$ the augmented process model, $\mathbf{w}^*_t \in \mathbb{R}^{n+m}$ is the augmented process noise term with a covariance $\mathbf{Q}^* = \mathrm{cov}(\mathbf{w}^*_t) \in \mathbb{R}^{(n+m)\times(n+m)}$, $\mathbf{H}^* \in \mathbb{R}^{m\times(n+m)}$ denotes the augmented observation model and $\mathbf{v}^*_t \in \mathbb{R}^m$ is the augmented process noise with a covariance $\mathbf{R}^* = \mathrm{cov}(\mathbf{w}^*_t) \in \mathbb{R}^{m\times m}$. The noise covariances are as follows:

$$\mathbf{Q}^* \;=\; \begin{bmatrix} \mathbf{Q} & 0 \\ 0 & \mathbf{\Sigma} \end{bmatrix} \tag{3.32}$$

$$\mathbf{R}^* \;=\; 0 \tag{3.33}$$

where $\mathbf{Q} = \mathrm{cov}(\mathbf{w}_t)$ and $\mathbf{\Sigma} = \mathrm{cov}(\varepsilon_t)$ for every $t \in \mathbb{N}$. As such, the state of the augmented system can be estimated using the standard Kalman filter, or any of its variants described in

Chapter 2, except the Information filter from Definition 2.5 because the necessary term $(\mathbf{R}^*)^{-1}$ cannot be evaluated.

In the context of the EKF, the UKF and the EnKF, the non-linear functions $f^*\colon \mathbb{R}^{n+m} \to \mathbb{R}^{n+m}$ and $h^*\colon \mathbb{R}^{n+m} \to \mathbb{R}^m$ corresponding to the augmented process model $\mathbf{F}^*$ and the augmented observation model $\mathbf{H}^*$, respectively, can be defined as:

$$f^*(\mathbf{x}, \mathbf{v}) = \begin{bmatrix} f(\mathbf{x}) \\ \boldsymbol{\Psi}\mathbf{v} \end{bmatrix} \tag{3.34}$$

$$h^*(\mathbf{x}, \mathbf{v}) = h(\mathbf{x}) + \mathbf{v} \tag{3.35}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{v} \in \mathbb{R}^m$ are function parameters, and $f\colon \mathbb{R}^n \to \mathbb{R}^n$ and $h\colon \mathbb{R}^n \to \mathbb{R}^m$ are the original process model and observation model functions, respectively, as in Definition 2.6 of the non-linear discrete-time dynamical system model. The above augmentation procedure is not to be confused with the similar augmentation procedure performed by the UKF (see Definition 2.9); both of them need to be performed independently.

Although the augmentation of the system state vector represents a very straightforward method to support a coloured observation noise, it has two important drawbacks. First, the number of state-space dimensions of the system increases to $n + m$, and therefore, the augmentation increases the computational complexity of the filter. Second, the measurements in the augmented system are perfect, i.e. contain no noise, which can lead to a numerical instability in the update step. The Kalman gain in the augmented system contains a matrix inversion term $\left[\mathbf{H}^* \mathbf{P}_t^* \mathbf{H}^{*\mathrm{T}} + \mathbf{R}^*\right]^{-1}$, where $\mathbf{P}_t^*$ denotes the *a priori* augmented state estimate covariance (see Definition 2.4). Because $\mathbf{R}^* = 0$, the matrix $\mathbf{H}^* \mathbf{P}_t^* \mathbf{H}^{*\mathrm{T}}$ is required to be invertible, and if it is too "close" to a singular matrix, the inversion will become ill-conditioned and numerically unstable on computers with a limited floating-point arithmetic precision. For example, this will happen if one of the variables of the *a priori* state estimate is known too precisely compared to the other variables, and thus the corresponding diagonal term in the covariance matrix $\mathbf{P}_t^*$ is too close to zero, compared to the others.

A detailed analysis of the problem of the potentially ill-conditioned inversion in the Kalman gain, as well as a discussion of possible algebraic remedies, can be found in (Maybeck, 1979, p. 249). As a workaround to the problem, (Wang et al., 2012) proposes two methods that modify the $\mathbf{H}^* \mathbf{P}_t^* \mathbf{H}^{*\mathrm{T}}$ matrix to ensure it is invertible in a numerically stable fashion: either using the Tikhonov regularisation, leading to the so called *Tikhonov KF algorithm*, or simply by adding a small coefficient to the diagonal elements of the matrix $\mathbf{P}_t^*$, leading to the so called

perturbed-P algorithm. Unfortunately, the first method is time consuming and might lead to a large regularisation error, while the second method is sensitive to the selection of the additive coefficient - if the coefficient is too large, the filter might become over-pessimistic and/or diverge. Overall, the problems of state augmentation led to the development of an alternative approach to accommodate the coloured observation noise, which is described in the next section.

3.3.2 Measurement Differencing

The *measurement differencing* approach, originally proposed in (Bryson and Henrikson, 1967) and sometimes referred to as *pre-whitening* or *time-differencing*, is another approach that enables the Kalman filter to accommodate a coloured observation noise. However, unlike state augmentation, it does not increase the number of state-space dimensions and it is numerically stable. In principle, measurement differencing rearranges the Kalman filter equations in a way that transforms the sequential correlation in the observation noise to a cross-correlation between the process and observation noises, allowing the estimation to be performed using one of the Kalman filter generalisations described in Section 3.1. Assume a system modelled as a linear discrete-time system from Definition 2.1:

$$\mathbf{x}_t \;=\; \mathbf{F}\,\mathbf{x}_{t-1} + \mathbf{w}_t \tag{3.36}$$

$$\mathbf{z}_t \;=\; \mathbf{H}\,\mathbf{x}_t + \mathbf{v}_t \tag{3.37}$$

with a coloured observation noise sequence $\{\mathbf{v}_t\}$ as in Definition 3.3:

$$\mathbf{v}_t \;=\; \mathbf{\Psi}\,\mathbf{v}_{t-1} \;+\; \varepsilon_{t-1} \tag{3.38}$$

where $\mathbf{\Psi} \in \mathbb{R}^{m\times m}$ and $\varepsilon_{t-1} \in \mathbb{R}^m$. The key idea of measurement differencing is to define a pseudo-measurement $\mathbf{z}^*_{t-1} \in \mathbb{R}^m$ as:

$$\mathbf{z}^*_{t-1} \;=\; \mathbf{z}_t - \mathbf{\Psi}\,\mathbf{z}_{t-1} \tag{3.39}$$

which can be rewritten (Simon, 2006, p. 190) using Equations (3.36)-(3.38) to:

$$\mathbf{z}^*_{t-1} \;=\; (\mathbf{H}\,\mathbf{F} - \mathbf{\Psi}\,\mathbf{H})\;\mathbf{x}_{t-1} \;+\; (\mathbf{H}\,\mathbf{w}_t \;+\; \varepsilon_{t-1}) \tag{3.40}$$

$$\;=\; \mathbf{H}^*\,\mathbf{x}_{t-1} \;+\; \mathbf{v}^*_{t-1} \tag{3.41}$$

Note that the pseudo-measurement noise sequence $\{\mathbf{v}_t^*\}$ is actually actually a white noise. This leads to a modified system model:

$$\mathbf{x}_t = \mathbf{F}\,\mathbf{x}_{t-1} + \mathbf{w}_t \tag{3.42}$$

$$\mathbf{z}_t^* = \mathbf{H}^*\,\mathbf{x}_t + \mathbf{v}_t^* \tag{3.43}$$

where $\mathbf{H}^* \in \mathbb{R}^{m\times n}$ represents a modified observation model and $\mathbf{v}_t^* \in \mathbb{R}^m$ a modified observation noise. The noise sequences $\{\mathbf{w}_t\}$, $\{\varepsilon_t\}$ and $\{\mathbf{v}_t^*\}$ are all white and have the following covariances:

$$\operatorname{cov}(\,\mathbf{w}_t\,) = \mathbf{Q} \tag{3.44}$$

$$\operatorname{cov}(\,\varepsilon_t\,) = \mathbf{\Sigma} \tag{3.45}$$

$$\operatorname{cov}(\,\mathbf{v}_t^*\,) = \mathbf{H}\,\mathbf{Q}\,\mathbf{H}^{\mathrm{T}} + \mathbf{\Sigma} \tag{3.46}$$

$$\operatorname{cov}(\,\mathbf{w}_{t+1}, \mathbf{v}_t^*\,) = \mathbf{Q}\,\mathbf{H}^{\mathrm{T}} \tag{3.47}$$

Due to the non-zero cross-covariance $\operatorname{cov}(\mathbf{w}_{t+1}, \mathbf{v}_t^*)$, the state of the system must be estimated using the Kalman filter generalisation for the type II correlation between the process and observation noise, which is described in Definition 3.2. Additionally, special care needs to be taken to correctly initialise the filter, as described (Bryson and Henrikson, 1967). Note that measurement differencing can be easily applied in the context of the EKF, the UKF and the EnKF; the observation model function $h^*\colon \mathbb{R}^n \to \mathbb{R}^m$ corresponding to the modified observation model $\mathbf{H}^*$ is defined as:

$$h^*(\mathbf{x}) = h(f(\mathbf{x})) + \mathbf{\Psi} h(\mathbf{x}) \tag{3.48}$$

where $\mathbf{x} \in \mathbb{R}^n$ is a function parameter and $f\colon \mathbb{R}^n \to \mathbb{R}^n$ is the original process model function as in Definition 2.6 of the non-linear discrete-time dynamical system model.

One of the limitations of the original Bryson and Henrikson's measurement differencing approach described above is that the observations $\{\mathbf{z}_t\}$, the coloured observation noise model parameter $\mathbf{\Psi}$ and the sequence $\{\varepsilon_t\}$ are all assumed to have the same number of dimensions m. (Gazit, 1997) provides an extended measurement differencing approach that eliminates such a restriction.

Another problem of the approach is that the pseudo-measurement $\mathbf{z}_t^*$ is computed from a future observation $\mathbf{z}_{t+1}$ (see Equation (3.39)), therefore the estimate produced by the update step at a time t represents a best estimate of the system state given all the measurements $\mathbf{z}_1, \ldots, \mathbf{z}_{t+1}$ up to time $t+1$. As such, the filter is effectively a lag-one smoother providing estimates with a latency of one time step. This latency can be problematic in certain applications, such as

in real-time control systems, and therefore, several modifications of the original measurement differencing approach have been proposed to eliminate it. For example, (Minkler and Minkler, 1993) provides a reinterpretation of the meaning of some of the filter parameters to alleviate the problem. (Rogers, 1987) developed an entirely different measurement differencing approach in the context of the alpha-beta filter (a simplified Kalman filter), which was later extended to the Kalman filter in (Guu and Wei, 1991) and also independently proposed in (Petovello et al., 2009). The main idea of this alternative approach is to define the pseudo-measurement $\mathbf{z}_t^*$ using only the past measurements as:

$$\mathbf{z}_t^* \;=\; \mathbf{z}_t - \boldsymbol{\Psi}\,\mathbf{z}_{t-1} \tag{3.49}$$

which leads to a new system model:

$$\mathbf{x}_t \;=\; \mathbf{F}\,\mathbf{x}_{t-1} + \mathbf{w}_t \tag{3.50}$$

$$\mathbf{z}_t^* \;=\; \left(\mathbf{H} \;-\; \boldsymbol{\Psi}\,\mathbf{H}\,\mathbf{F}^{-1}\right)\,\mathbf{x}_t + \left(\boldsymbol{\Psi}\,\mathbf{H}\,\mathbf{F}^{-1}\,\mathbf{w}_t \;+\; \varepsilon_{t-1}\right) \tag{3.51}$$

$$\;=\; \mathbf{H}^*\,\mathbf{x}_t + \mathbf{v}_t^* \tag{3.52}$$

where the noise sequences $\{\mathbf{w}_t\}$, $\{\varepsilon_t\}$ and $\{\mathbf{v}_t^*\}$ are all white and have the following covariances:

$$\mathrm{cov}(\,\mathbf{w}_t\,) \;=\; \mathbf{Q} \tag{3.53}$$

$$\mathrm{cov}(\,\varepsilon_t\,) \;=\; \boldsymbol{\Sigma} \tag{3.54}$$

$$\mathrm{cov}(\,\mathbf{v}_t^*\,) \;=\; \boldsymbol{\Psi}\,\mathbf{H}\,\mathbf{F}^{-1}\,\mathbf{Q}\,\left(\mathbf{F}^{-1}\right)^{\mathrm{T}}\,\mathbf{H}^{\mathrm{T}}\,\boldsymbol{\Psi}^{\mathrm{T}} + \boldsymbol{\Sigma} \tag{3.55}$$

$$\mathrm{cov}(\,\mathbf{w}_t, \mathbf{v}_t^*\,) \;=\; \mathbf{Q}\,\left(\mathbf{F}^{-1}\right)^{\mathrm{T}}\,\mathbf{H}^{\mathrm{T}}\,\boldsymbol{\Psi}^{\mathrm{T}} \tag{3.56}$$

Due to presence of a non-zero cross-covariance $\mathrm{cov}(\mathbf{w}_t, \mathbf{v}_t^*)$, the system state must be estimated using the Kalman filter generalisation, this time for the type I correlation between the process and observation noise, as described in Definition 3.1.

The main drawback of the alternative measurement differencing approach described here is the requirement to compute the inverse process model $\mathbf{F}^{-1}$. Although such an inverse is guaranteed to exist (Gelb, 1974), it might also be ill-conditioned (Wang et al., 2012), and moreover, if the process model is time-varying, such as in the EKF where it is computed from a Jacobian of a non-linear function f (see Section 2.4), the inversion will significantly increase the computational cost of the update step. Also, the whole approach is practically unusable in the context of the UKF and the EnKF, firstly, because both these filters were built on the premise to avoid the computation of Jacobians (assuming an inverse process model function f^{-1} is not easily available), and secondly, because there are no equivalent generalisations of these filters for the type I correlation between the process and observation noise, as discussed in Section 3.1.

3.4 Combined and Higher-Order Correlations

The previous sections describe the most important variants of a first-order correlation in and between the process and observation noises encountered in Kalman filter applications, and review approaches that provide optimal state estimates in the presence of such correlations. In fact, the methods described cover the absolute majority of practical estimation problems involving correlated noises. Nevertheless, several other methods have been proposed to address certain more special cases of correlation.

(Jiang et al., 2010) describe a set of methods that support any finite-step sequential correlation in the process and observation noise sequences, as well as any finite-step correlation between the process and observation noise sequences. These methods are further extended in (Li et al., 2011) to allow for an arbitrary combination of such finite-step correlations. The main insight provided by these papers is that in order to support such higher-order correlations in a recursive fashion, the Kalman gain needs to be decomposed into two multiplicative components and each of them maintained independently. A similar method was developed earlier in (Li et al., 2000), but it provides optimal state estimates in a batch-mode only, i.e. not in a recursive filter. It should be noted that practical applicability of these approaches is somewhat limited, because higher-order models of correlations of real-world physical noise processes are rarely available.

3.5 Chapter Summary

This chapter reviewed modifications of the Kalman filter and its variants that enable them to operate in the presence of correlations between the state estimate error and the observation error, caused by the presence of a sequential correlation or mutual cross-correlations in the process and observation noises. The number of recent publications indicates that this is still an active area of research, even more than 50 years since the development of the Kalman filter. In general, all the methods discussed expect well-defined Gauss-Markov models of the underlying physical noise processes. Unfortunately, in certain applications, such models might be to prohibitive, characterise the true noise process inaccurately, and therefore affect the quality of the state estimates computed by the filter.

With the new EnKF-based estimation methods proposed in Chapter 5 of this thesis, there is no restriction on the mathematical model of the noise processes and correlations, the only requirement is the ability to emulate the effects of the noise process on the random samples

(ensembles) representing the state estimates and the observations. Additionally, these new methods are useful in situations where the conventional approaches presented in this chapter, such as state augmentation, would incur an unacceptable performance penalty, as discussed in Section 3.2.

Chapter 4

The Common Past

This thesis deals with the general problem of a linear state estimation in the presence of a correlation between the state estimate error and the observation error. Chapter 3 reviews extensions of the Kalman filter applicable in situations where such a correlation is caused by a sequential correlation in the observation noise sequence or by a cross-correlation between the process and observation noise sequences. However, this correlation can also be caused by the presence of *common past information* shared between the state estimate and the observation, which results from a situation where information derived from a single sensor measurement has a chance to affect the state estimate at multiple times. In the literature, this problem is also referred to, less poetically, as *data incest*, as well as *double-counting*, *common information* or, in particular in the context of information theory, as *mutual information*. The principal difference from the case of the correlated noises is that in the presence of common past information it is generally difficult to algebraically express the relation (i.e. the correlation) between the state estimate and the observation errors, which is a necessary prerequisite to perform the estimation in a consistent and optimal fashion. This chapter reviews approaches that attempt to do exactly that.

As briefly discussed in Chapter 1, the common past information problem is characteristic of sensor networks - systems in which observations of physical phenomena are performed by a number of communicating agents (Akyildiz et al., 2002). For example, sensor networks are practical in military applications, such as tracking of moving targets using sensors deployed in multiple geographical locations. Formally, a sensor network is defined as follows:

Definition 4.1 (Sensor network). Assume a set of *agents* (aka *nodes*) $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$, each of them equipped with one or more sensors that observe some (possibly different) physical system.

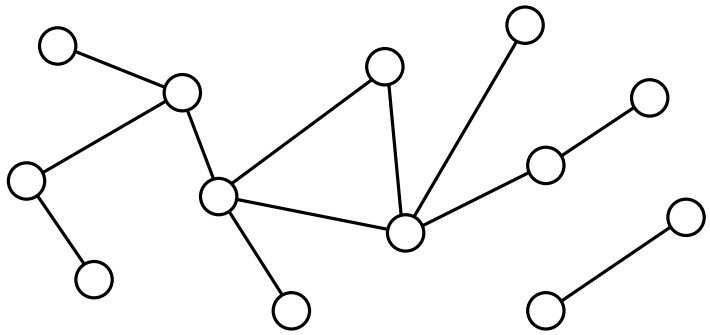


Fig. 4.1: Sensor network with a random topology.

At a time t, sensors of an agent $\mathcal{A}_i$ produce an observation vector $\mathbf{z}_i(t) \in \mathbb{R}^{m_i}$, which is affected by an observation noise term $\mathbf{v}_i(t) \in \mathbb{R}^{m_i}$ with a covariance $\mathbf{R}_i = \mathrm{cov}(\mathbf{v}_i(t)) \in \mathbb{R}^{m_i \times m_i}$, for every $i = 1, \ldots, k$ and $t \in \mathbb{N}$. The observation noise is assumed white, Gaussian, and independent.

Additionally, the agents can exchange information with each other. Denote the communication network available to the agents at a time t as a set $\mathcal{E}(t) \subseteq \{\{a, b\} \mid a, b \in \mathcal{A}\}$: if there is a direct communication link between agents $\mathcal{A}_i$ and $\mathcal{A}_j$, then $\{\mathcal{A}_i, \mathcal{A}_j\} \in \mathcal{E}(t)$. Note that the communication links are assumed bi-directional. Let the symbol $\mathcal{N}_i(t)$ denote the set of indices of agents (*neighbours*) that can send a message to the agent $\mathcal{A}_i$ at a time t, formally $\mathcal{N}_i(t) = \{\, j \mid \{\mathcal{A}_i, \mathcal{A}_j\} \in \mathcal{E}(t) \;\&\; i \neq j \,\}$.

The sets $\mathcal{A}$ and $\mathcal{E}(t)$ effectively represent a set of vertices and edges, respectively, of an undirected graph $\mathcal{G}(t) = (\mathcal{A}, \mathcal{E}(t))$. A sequence $\{\mathcal{G}(t)\}$ of such graphs with time-varying edges will be called a *sensor network* (or *switching network*). If the edges of the sensor network are not changing over time (*non-switching network*), the time parameter can be dropped and the sensor network will be denoted simply as $\mathcal{G} = (\mathcal{A}, \mathcal{E})$, and similarly, the set of neighbours denoted as $\mathcal{N}_i$. An example of a sensor network is depicted in Figure 4.1.

This chapter reviews state-of-the-art approaches to linear state estimation based on observations obtained by a sensor network, which is an instance of a more general *data fusion* problem. The approaches will be presented in order from centralised to distributed ones, which best illustrates the nature of the common past information problem, and its treatment. For clarity, this chapter assumes a slightly different notation of variables than the previous chapters: the time index is written as a parameter (e.g. $\mathbf{x}(t)$) instead of a subscript ($\mathbf{x}_t$), and the vacant subscript is designated to an index of a node in the sensor network to which the variable is related ($\mathbf{x}_i(t)$), wherever necessary. Note that such notation is a de facto standard in sensor network literature.

4.1 Centralised Kalman Filter

The most straightforward approach to estimate the state of a physical system from measurements obtained by a sensor network is to collect all the measurements in a single *fusion centre*, and perform the estimation there using an algorithm such as the Kalman filter. Assume a non-switching sensor network $\mathcal{G} = (\mathcal{A}, \mathcal{E})$ as in Definition 4.1, and assume the agent $\mathcal{A}_1$ represents a fusion centre such that all the other agents can directly communicate with it, i.e.

$$\forall\, i = 2, \ldots, k \quad : \quad \{\mathcal{A}_1, \mathcal{A}_i\} \in \mathcal{E} \tag{4.1}$$

At every time step $t \in \mathbb{N}$, all agents sends all the measurements to the fusion centre, where they are combined into a single measurements vector $\mathbf{z}(t) \in \mathbb{R}^m$, affected by a combined observation noise $\mathbf{v}(t) \in \mathbb{R}^m$ with a covariance $\mathbf{R} = \mathrm{cov}(\mathbf{v}(t)) \in \mathbb{R}^{m \times m}$, all defined as:

$$\mathbf{z}(t) = \begin{bmatrix} \mathbf{z}_1(t) \\ \vdots \\ \mathbf{z}_k(t) \end{bmatrix} \qquad \mathbf{v}(t) = \begin{bmatrix} \mathbf{v}_1(t) \\ \vdots \\ \mathbf{v}_k(t) \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{R}_k \end{bmatrix} \tag{4.2}$$

where $m = \sum_{i=1}^{k} m_i$. The physical system is modelled using the standard linear discrete-time system model from Definition 2.1:

$$\mathbf{x}(t) = \mathbf{F}\,\mathbf{x}(t-1) + \mathbf{w}(t) \tag{4.3}$$

$$\mathbf{z}(t) = \mathbf{H}\,\mathbf{x}(t) + \mathbf{v}(t) \tag{4.4}$$

with a suitable $\mathbf{F} \in \mathbb{R}^{n \times n}$ and $\mathbf{H} \in \mathbb{R}^{m \times n}$. Because all the necessary assumptions are satisfied, the state of such a system can be estimated using the traditional Kalman filter, or any of its variants, as described in Chapter 2. An example of a sensor network with a topology required by the centralised Kalman filter is provided in Figure 4.2.

Although the centralised Kalman filtering approach described above provides optimal estimates and it is very simple to understand and implement, it has several principal drawbacks (Durrant-Whyte et al., 2001). First, the fusion centre represents a single point of failure of the whole sensor network; if the agent $\mathcal{A}_1$ fails or loses a communication link from the other agents, data fusion in the network halts. Second, the centralised approach does not scale to large sensor networks, due to a limited communication bandwidth and computational resources available to the fusion centre. Third, such an all-to-one communication pattern requires a certain global knowledge of the network topology and a complete connectivity in the network at

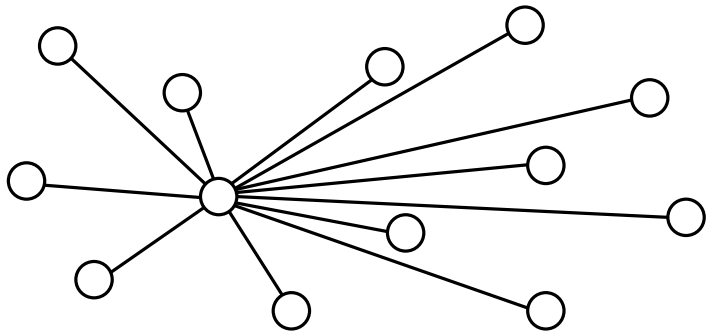


Fig. 4.2: Sensor network in the centralised Kalman filter.

all times, which is difficult to achieve in many practical applications. These problems served as a motivation for a development of several decentralised approaches to data fusion in sensor networks, which are described in the following sections. The centralised approach serves well as a benchmark in discussions of the quality of the estimates provided by other approaches.

4.2 Decentralised Kalman Filter

The issues associated with a presence of a single fusion centre in a sensor network can be eliminated by decentralising the Kalman filter computation among all the nodes. This approach was first proposed in (Durrant-Whyte et al., 1990), and it is referred to as the *Decentralised Kalman filter* (DKF). The basic idea behind the DKF is that each node in the sensor network maintains its own local estimate of the state of the physical system, not just the fusion centre. At every time step, each node broadcasts its local observation to all the other nodes in the network, and upon receiving such information from the other nodes, it is able to construct an optimal global state estimate, which is numerically equivalent to a would-be estimate of a centralised filter.

Formally, assume a non-switching sensor network $\mathcal{G} = (\mathcal{A}, \mathcal{E})$ from Definition 4.1, which is fully-connected, i.e.

$$\forall\, i, j \in \{1, \ldots, k\} \quad : \quad \{\mathcal{A}_i, \mathcal{A}_j\} \in \mathcal{E} \tag{4.5}$$

Each node $\mathcal{A}_i$ models the global physical system locally as:

$$\mathbf{x}(t) \;=\; \mathbf{F}\,\mathbf{x}(t-1) + \mathbf{w}(t) \tag{4.6}$$

$$\mathbf{z}_i(t) \;=\; \mathbf{H}_i\,\mathbf{x}(t) + \mathbf{v}_i(t) \tag{4.7}$$

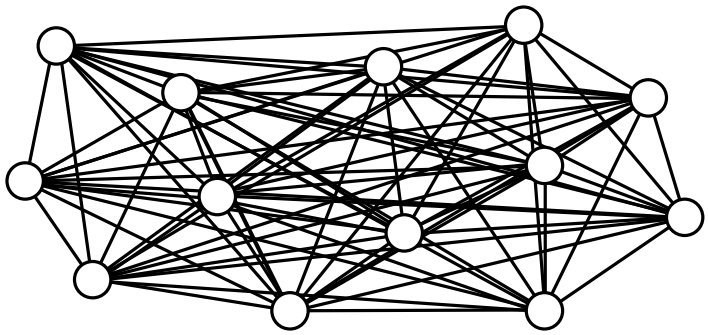


Fig. 4.3: Sensor network in the Decentralised Kalman filter.

where $\mathbf{x}(t) \in \mathbb{R}^n$ denotes the system state vector at a time t, $\mathbf{F} \in \mathbb{R}^{n \times n}$ is the process model, and $\mathbf{w}(t) \in \mathbb{R}^n$ is the process noise with a covariance $\mathbf{Q} = \mathrm{cov}(\mathbf{w}(t)) \in \mathbb{R}^{n \times n}$. The local observation vector $\mathbf{z}_i(t) \in \mathbb{R}^{m_i}$ is affected by an observation noise $\mathbf{v}_i(t) \in \mathbb{R}^{m_i}$ with a covariance $\mathbf{R}_i = \mathrm{cov}(\mathbf{v}_i(t)) \in \mathbb{R}^{m_i \times m_i}$, and its relation to the system state is given by a local observation model $\mathbf{H}_i \in \mathbb{R}^{m_i \times n}$. An equivalent global system model would be defined as:

$$\mathbf{x}(t) \;=\; \mathbf{F}\,\mathbf{x}(t-1) + \mathbf{w}(t) \tag{4.8}$$

$$\mathbf{z}(t) \;=\; \mathbf{H}\,\mathbf{x}(t) + \mathbf{v}(t) \tag{4.9}$$

where the observation vector $\mathbf{z}(t) \in \mathbb{R}^m$, the observation model $\mathbf{H} \in \mathbb{R}^{m \times n}$, the observation noise $\mathbf{v}(t) \in \mathbb{R}^m$, and the observation noise covariance $\mathbf{R} = \mathrm{cov}(\mathbf{v}(t)) \in \mathbb{R}^{m \times m}$, respectively, are composed of the local ones as:

$$\mathbf{z}(t) \;=\; \begin{bmatrix} \mathbf{z}_1(t) \\ \vdots \\ \mathbf{z}_k(t) \end{bmatrix} \quad \mathbf{H} \;=\; \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_k \end{bmatrix} \quad \mathbf{v}(t) \;=\; \begin{bmatrix} \mathbf{v}_1(t) \\ \vdots \\ \mathbf{v}_k(t) \end{bmatrix} \quad \mathbf{R} \;=\; \begin{bmatrix} \mathbf{R}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{R}_k \end{bmatrix} \tag{4.10}$$

and $m = \sum_{i=1}^{k} m_i$. An example of a sensor network with a topology required by the Decentralised Kalman filter is provided in Figure 4.3.

The DKF estimates the state locally at every node using the Information filter, which enables the nodes to assimilate observations from the other nodes into the local state estimate simply by summation, as discussed in Section 2.3. Nevertheless, the use of the Information filter is a mathematical convenience rather than a necessity; optimal estimates can also be obtained by the conventional Kalman filter, although with a higher computational cost. Each node $\mathcal{A}_i$ represents its state estimate at a time t using an information vector $\hat{\mathbf{y}}_i(t) \in \mathbb{R}^n$ and an information matrix

$\mathbf{Y}_i(t) \in \mathbb{R}^{n \times n}$, respectively, defined as:

$$\hat{\mathbf{y}}_i(t) \;=\; \mathbf{P}_i^{-1}(t)\,\hat{\mathbf{x}}_i(t) \tag{4.11}$$

$$\mathbf{Y}_i(t) \;=\; \mathbf{P}_i^{-1}(t) \tag{4.12}$$

where $\hat{\mathbf{x}}_i(t) \in \mathbb{R}^n$ denotes a state estimate vector and $\mathbf{P}_i(t) \in \mathbb{R}^{n \times n}$ denotes the associated error covariance matrix. In order to maintain the local state estimate in such a representation, the node performs recursively the following steps:

1. **Prediction**

   Given a local state estimate from the previous time step $t-1$, represented using an information vector $\hat{\mathbf{y}}_i(t-1)$ and an information matrix $\mathbf{Y}_i(t-1)$, compute the *a priori* state estimate at a time t, represented using an information vector $\hat{\mathbf{y}}_i^-(t)$ and information matrix $\mathbf{Y}_i^-(t)$. This can be done either using the Information filter's prediction step as in Equations (2.27)-(2.30), or using the traditional Kalman filter prediction step with an information form conversion as in Equations (2.37)-(2.38), i.e.

   $$\hat{\mathbf{y}}_i^-(t) \;=\; \mathbf{Y}_i^-(t)\,\mathbf{F}\,\mathbf{Y}_i^{-1}(t-1)\,\hat{\mathbf{y}}_i(t-1) \tag{4.13}$$

   $$\mathbf{Y}_i^-(t) \;=\; \left[\,\mathbf{F}\,\mathbf{Y}_i^{-1}(t-1)\,\mathbf{F}^{\mathrm{T}} \;+\; \mathbf{Q}\,\right]^{-1} \tag{4.14}$$

2. **Observation broadcast**

   Send local observation information terms $[\mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{z}_i(t)]$ and $[\mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{H}_i]$ (also called *state error info* and *variance error info*, respectively) to all the nodes in the network.

3. **Update**

   Compute the *a posteriori* state estimate by assimilating all the local observation information terms received from the other nodes in the network, using the Information filter update rule:

   $$\hat{\mathbf{y}}_i(t) \;=\; \hat{\mathbf{y}}_i^-(t) \;+\; \sum_{j=1}^{k} \mathbf{H}_j^{\mathrm{T}}\,\mathbf{R}_j^{-1}\,\mathbf{z}_j(t) \tag{4.15}$$

   $$\mathbf{Y}_i(t) \;=\; \mathbf{Y}_i^-(t) + \sum_{j=1}^{k} \mathbf{H}_j^{\mathrm{T}}\,\mathbf{R}_j^{-1}\,\mathbf{H}_j \tag{4.16}$$

   The result represents a globally optimal state estimate, given all the observations up to time t from all the sensor nodes.

The basic DKF approach described above can be modified in a number of interesting ways. For example, as suggested in (Durrant-Whyte et al., 1990), the nodes typically do not need to model the full system state, but only a locally relevant part thereof, which enables the nodes to only communicate appropriate parts of the observation information terms and thus save communicational and computational resources. Consequently, the estimates can potentially be computed faster in the DKF than in the case of the centralised Kalman filter, because each of the nodes operates in parallel with potentially lower-dimensional matrices and vectors than a fusion centre would. Furthermore, the DKF can be modified to enable the nodes to operate asynchronously instead of synchronous time steps, which improves its applicability in real-world data fusion problems (Rao et al., 1993). Also, as shown in (Grime and Durrant-Whyte, 1994), the centralised and the decentralised Kalman filter approaches can be combined within a single sensor network, for example, the sensor network can be organised hierarchically so that local groups of nodes perform centralised estimation, and the decentralised estimation is only applied among the local fusion centres.

Although the DKF eliminates the risk of a single-point failure inherent to the centralised Kalman filter approach, it does not address the issue of scalability to a large number of sensor nodes; in fact, the all-to-all communication pattern in the DKF is even more demanding on the communication resources than the all-to-one pattern of the centralised approach. If the optimality of estimates is not required, the DKF algorithm can be easily modified to limit the raw measurement exchanges to nearby nodes only, e.g. to the 2-hop neighbourhood of the source node (some-to-some communication pattern), and thus achieve the scalability. However, in such a network, the accuracy of the local estimates might degrade to a level that is not acceptable for practical applications, because sensor measurements will only have a limited spatial effect (Grime et al., 1992).

4.3 Communication of Local Estimates

In order to increase the spatial reach of the sensor measurements and reduce the amount of data communicated, the nodes might opt to communicate their local estimates instead of the raw sensor measurements (i.e. transmit $\hat{\mathbf{y}}_i^-(t)$ and $\mathbf{Y}_i^-(t)$ instead of $[\mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{z}_i(t)]$ and $[\mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{H}_i]$). Such a modification potentially leads to a truly distributed and scalable sensor fusion architecture, which can be characterised as (Spanos and Murray, 2005):

- The network can have an arbitrary and dynamically changing topology.

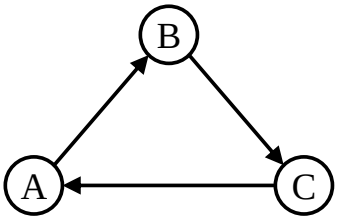


Fig. 4.4: Propagation of common past information in a sensor network.

- All the nodes maintain an optimal state estimate, there are no specialised fusion centres.
- The nodes only exchange a bounded number of messages with their local neighbours, "flooding" of the network is not acceptable.

Unfortunately, a sensor network communicating local estimates instead of raw measurements will become exposed to the common past information problem, which is best illustrated with an example. Assume a fully-connected sensor network composed of three nodes, as depicted in Figure 4.4, where each node estimates the state of a single physical process using its own sensor measurements and using estimates communicated from the other nodes, which are also considered as measurements in the update step of a node's filter. Suppose node A sent its estimate to node B, which used it to update its own estimate and subsequently sent this estimate to node C, which also updated its own estimate. Now, if node C sends its estimate back to node A, the node A cannot simply consider it as an independent measurement, because C's estimate is statistically dependent and correlated to A's estimate - they share common past information. If the node A ignored this correlation and used a conventional Kalman filter update rule, its estimate would become overconfident, in the sense that the associated error covariance will be "smaller" than the covariance of the true error, as formally defined in Section 2.2. Furthermore, the problem would worsen with every subsequent repetition of such a communication cycle, and eventually all the local estimates would converge to a wrong value even though potentially no new observations were made.

The common past information problem is not limited to a situation where the nodes estimate the state of the same physical process. As discussed in the previous section, for efficiency reasons each node is motivated to only model as small a part of the system state as possible, typically using only state variables that directly affect the node's actions and variables required by the observation model of the node's sensors. Unfortunately, in certain applications, this comes at a price that certain sensor measurements cannot be correctly interpreted given the node's system

model. The node then faces two options: either to ignore such measurements altogether and thus degrade its estimation performance, or to reinterpret the observation using a system model and an estimate maintained by another node. For example, in cooperative localisation, a vehicle can only correctly interpret a measurement of its relative distance to a neighbouring vehicle if it has a model of the neighbouring vehicle's position. However, the vehicle can reinterpret the distance measurement using a position estimate maintained by the neighbour vehicle, and thus compute a virtual measurement indicating its own position. Unfortunately, the neighbouring vehicle might have done the same thing in the past, and therefore such a virtual observation might be already correlated to the local state estimate - once again, the nodes share common past information.

Note that a node cannot easily augment its state estimate vector with new variables corresponding to the previously unmodelled state variables that are communicated by another node, because the cross-correlation between the "old" and the "new" state variables is generally not known and therefore, the augmented state estimate's covariance matrix cannot be reconstructed in general. The reinterpretation of the measurements discussed above is actually equivalent to assuming that such a cross-correlation is zero, which leads to the same common past information problem. Generally speaking, there is an inherent trade-off between the scalability/flexibility of a sensor network and the quality of the estimates computed therein: a sensor network can be extremely scalable and flexible, but the estimates will either be inconsistent due to the common past information problem, or they will have a poor accuracy due to the limited spatial reach of the sensor measurements. The data fusion approaches discussed in the following sections represent various compromises to such a trade-off.

4.4 Channel Filters

In order to perform a consistent state estimation in a sensor network where the nodes communicate estimates instead of raw sensor measurements, the nodes need to have some means to eliminate common past information from the data received from other nodes. In tree-connected networks, there is just one communication path between any two nodes, and therefore it is possible to keep track of common past information shared between every pair of neighbouring nodes simply by monitoring information exchanged through the communication channel between the two, using a so-called *Channel filter*, first introduced in (Grime et al., 1992). Whenever a node receives a state estimate from a neighbouring node, the common past information maintained

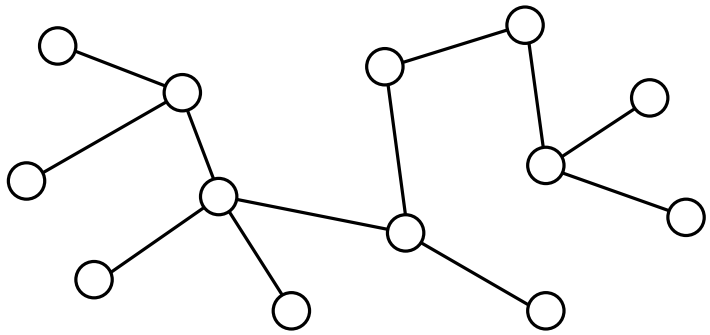


Fig. 4.5: Sensor network in the Channel filters approach.

by the Channel filter corresponding to the communication channel is eliminated from the communicated estimate, and thus the estimate can then be regarded as a measurement affected by a white noise. Using this approach, each node can maintain a globally optimal and consistent state estimate, equivalent to a would-be centralised estimate, up to an inherent communication delay. With the information form of the local filter equations, the resulting data fusion system is surprisingly simple. An example of a sensor network with a topology required by the Channel filters approach is provided in Figure 4.5.

Formally, assume a non-switching sensor network $\mathcal{G} = (\mathcal{A}, \mathcal{E})$ as in Definition 4.1, and assume that the network is tree-connected, i.e. there exists exactly one simple (without repeating nodes) undirected path between any two nodes. As in the case of the Decentralised Kalman filter described in Section 4.2, each node $\mathcal{A}_i$ models the physical system locally as:

$$\mathbf{x}(t) \;=\; \mathbf{F}\,\mathbf{x}(t-1) + \mathbf{w}(t) \tag{4.17}$$

$$\mathbf{z}_i(t) \;=\; \mathbf{H}_i\,\mathbf{x}(t) + \mathbf{v}_i(t) \tag{4.18}$$

and it represents an estimate of the system state at a time t using an information vector $\hat{\mathbf{y}}_i(t) \in \mathbb{R}^n$ and an information matrix $\mathbf{Y}_i(t) \in \mathbb{R}^{n\times n}$. In the Channel filters approach the node performs recursively the following steps in order to maintain such an estimate:

1. **Prediction**

   Given a local state estimate from the previous time step $t-1$, represented using an information vector $\hat{\mathbf{y}}_i(t-1)$ and an information matrix $\mathbf{Y}_i(t-1)$, compute the *a priori* state estimate, represented as an information vector $\hat{\mathbf{y}}_i^-(t) \in \mathbb{R}^n$ and information matrix

$\mathbf{Y}_i^-(t) \in \mathbb{R}^{n \times n}$, using the standard Kalman filter prediction rule:

$$\hat{\mathbf{y}}_i^-(t) \;=\; \mathbf{Y}_i^-(t)\,\mathbf{F}\,\mathbf{Y}_i^{-1}(t-1)\,\hat{\mathbf{y}}_i(t-1) \tag{4.19}$$

$$\mathbf{Y}_i^-(t) \;=\; \left[\,\mathbf{F}\,\mathbf{Y}_i^{-1}(t-1)\,\mathbf{F}^{\mathrm{T}} \;+\; \mathbf{Q}\,\right]^{-1} \tag{4.20}$$

2. **Local update**

   Update the *a priori* estimate using a local observation $\mathbf{z}_i(t) \in \mathbb{R}^{m_i}$ to obtain a *partial estimate* using the Information filter update rule:

$$\tilde{\mathbf{y}}_i(t) \;=\; \hat{\mathbf{y}}_i^-(t) \,+\, \mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{z}_i(t) \tag{4.21}$$

$$\tilde{\mathbf{Y}}_i(t) \;=\; \mathbf{Y}_i^-(t) + \mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{H}_i \tag{4.22}$$

3. **Channel data exchange**

   Suppose there is a neighbouring node $\mathcal{A}_j$, and the Channel filter corresponding to the communication channel between $\mathcal{A}_i$ and $\mathcal{A}_j$ has an estimate of the common past information shared between the two nodes at the previous time step $t-1$, represented using an information vector $\hat{\mathbf{y}}_{ij}(t-1) \in \mathbb{R}^n$ and an information matrix $\mathbf{Y}_{ij}(t-1) \in \mathbb{R}^{n \times n}$. The Channel filter first propagates the common past information estimate to the current time step, for example, using the standard Kalman filter prediction rule as:

$$\hat{\mathbf{y}}_{ij}^-(t) \;=\; \mathbf{Y}_{ij}^-(t)\,\mathbf{F}\,\mathbf{Y}_{ij}^{-1}(t-1)\,\hat{\mathbf{y}}_{ij}(t-1) \tag{4.23}$$

$$\mathbf{Y}_{ij}^-(t) \;=\; \left[\,\mathbf{F}\,\mathbf{Y}_{ij}^{-1}(t-1)\,\mathbf{F}^{\mathrm{T}} \;+\; \mathbf{Q}\,\right]^{-1} \tag{4.24}$$

   The nodes $\mathcal{A}_i$ and $\mathcal{A}_j$ then exchange their local partial estimates, and the Channel filter uses this data to update the common past information estimates as:

$$\hat{\mathbf{y}}_{ij}(t) \;=\; \tilde{\mathbf{y}}_i(t) \,+\, \tilde{\mathbf{y}}_j(t) \,-\, \hat{\mathbf{y}}_{ij}^-(t) \tag{4.25}$$

$$\mathbf{Y}_{ij}(t) \;=\; \tilde{\mathbf{Y}}_i(t) + \tilde{\mathbf{Y}}_j(t) - \mathbf{Y}_{ij}^-(t) \tag{4.26}$$

   Note that all the operations performed in this step are symmetric for both the nodes $\mathcal{A}_i$ and $\mathcal{A}_j$, and they need to be repeated for every neighbouring node.

4. **Final update**

   Compute the final *a posteriori* state estimate using all the partial estimates received from the neighbouring nodes, and the common past information estimates provided by the

respective Channel filters as:

$$\hat{\mathbf{y}}_i(t) \;=\; \tilde{\mathbf{y}}_i(t) \;+\; \sum_{j \in \mathcal{N}_i} \Big(\tilde{\mathbf{y}}_j(t) \;-\; \hat{\mathbf{y}}^{-}_{ij}(t-1) \Big) \tag{4.27}$$

$$\mathbf{Y}_i(t) \;=\; \tilde{\mathbf{Y}}_i(t) + \sum_{j \in \mathcal{N}_i} \Big(\tilde{\mathbf{Y}}_j(t) - \mathbf{Y}^{-}_{ij}(t-1) \Big) \tag{4.28}$$

where $\mathcal{N}_i$ denotes a set of neighbouring nodes' indices. The result represents a globally optimal state estimate, given all the observations from all the sensor nodes in the network that could have been received by the node $\mathcal{A}_i$ up to time t, considering the network topology and the inherent time delay caused by message forwarding.

The simplicity of the Channel filters' equations stems from the fact that assimilation of information terms into an estimate is an associative operation (addition). However, Channel filters are not at all limited to the Information filter or even Gaussian probability distributions, and the core principle can be applied with other probability distributions representations, for example, in the context of particle filters (Ong et al., 2006). Although the Channel filters algorithm described above assumes that all the nodes operate in synchronous time steps, the algorithm can be easily modified to a fully asynchronous operation (Grime and Durrant-Whyte, 1994).

As discussed in (Durrant-Whyte et al., 2001), Channel filters have two important characteristics. First, every node only communicates with each of its neighbours once per local sensor observation cycle and therefore, the number of messages sent by every node is bounded regardless of the network topology beyond the single-hop neighbourhood, enabling the network to scale indefinitely. Second, if a communication link between two nodes is temporarily broken, the Channel filter can suspend and resume its operation without any loss of data.

The main drawback of the Channel filters approach is the requirement for the time-invariant tree-connected network topology, which makes the approach unsuitable to dynamic environments and mobile nodes. Additionally, if a communication link between two nodes or a node itself fails indefinitely, the recovery of the data fusion operation without discarding the current estimates is a non-trivial problem. (Dodin and Nimier, 2002) presents a method addressing this problem, which effectively extends the Channel filters approach to networks where the number of paths between two nodes is greater than one.

4.5 Kalman Consensus Algorithms

The problem of reaching a common state estimate by nodes in a sensor network can be viewed as the well-known problem of reaching a consensus in a distributed system (Lynch, 1996). This problem can be illustrated using the following example (Ren et al., 2004). Assume a group of friends trying to schedule a dinner. The venue of the dinner is known in advance but the time is not, and as usual, every person has their own time preference. A centralised solution of this meet-for-dinner problem would be to make a conference call between all the people, and agree on a specific time. Unfortunately, the people only have phones that allow them to talk to a single person at a time, and also some people do not have each others' number. In this situation, the dinner can be scheduled by applying the *average consensus* strategy, in which each person repeatedly calls to several other people, and simply averages their time preferences, and uses that average as her new time preference. If all the people apply this strategy, and there are no complete strangers in the group, each person's time preference will eventually converge towards a single consensual value. Additionally, if some people have a stronger preference on the time of the dinner than others, the strategy can be modified to consider a weight of the preferences, leading to the so-called *weighted-average consensus* algorithm. This very strategy can be applied in a sensor network, enabling the nodes to reach a consensual state estimate. If the weight factors are based on the uncertainty (covariance) associated with the communicated estimates, such an algorithm is called a *Kalman Consensus algorithm*, or a *Distributed Kalman filter*. Generally speaking, it is rather straightforward to design a Kalman Consensus algorithm that guarantees, under certain assumptions, that all the nodes converge to a single state estimate. However, the algorithm also needs to guarantee that the local estimates are consistent, not affected by the common past information, and that they are close enough to the optimal would-be centralised state estimate, after a short enough time. These are the main concerns addressed by various Kalman Consensus algorithms that have been described in the literature, several of whom will be reviewed in the following sections. For clarity of explanation, the algorithms are split into two principal classes, depending whether or not the underlying dynamical system changes its state over time.

4.5.1 Static Systems

A static system is a special case of the linear discrete-time dynamical system from Definition 2.1, in which the system state $\mathbf{x} \in \mathbb{R}^n$ is constant over time. Moreover, the system is only observed

once by each of the nodes in the sensor network, a node $\mathcal{A}_i$ models such an observation as:

$$\mathbf{z}_i \;=\; \mathbf{H}_i\,\mathbf{x} + \mathbf{v}_i \tag{4.29}$$

where $\mathbf{z}_i \in \mathbb{R}^{m_i}$ denotes the sole observation, $\mathbf{H}_i \in \mathbb{R}^{m_i \times n}$ is an observation model, $\mathbf{v}_i \in \mathbb{R}^{m_i}$ is an observation noise with a covariance $\mathbf{R}_i = \mathrm{cov}(\mathbf{v}_i) \in \mathbb{R}^{m_i \times m_i}$, and m_i denotes the dimension of the local observation. The static system enables a simpler design of the Kalman Consensus algorithm than in the case of a general dynamical system. Similarly to other data fusion algorithms presented in this chapter, the Kalman Consensus algorithms can be conveniently described in the context of the Information filter. Suppose the state estimate available to a node $\mathcal{A}_i$ at a time t is represented using a so-called *composite* information vector $\hat{\mathbf{y}}_i(t) \in \mathbb{R}^n$ and a composite information matrix $\mathbf{Y}_i(t) \in \mathbb{R}^{n \times n}$ for $i = 1, \ldots, k$. At time $t = 0$, the node initialises the estimate using the sole observation as:

$$\hat{\mathbf{y}}_i(0) \;=\; \mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{z}_i \tag{4.30}$$

$$\mathbf{Y}_i(0) \;=\; \mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{H}_i \tag{4.31}$$

The Kalman Consensus algorithm performs recursively the following computation:

$$\hat{\mathbf{y}}_i(t+1) \;=\; W_{ii}(t)\,\hat{\mathbf{y}}_i(t) + \sum_{j \in \mathcal{N}_i(t)} W_{ij}(t)\,\hat{\mathbf{y}}_j(t) \tag{4.32}$$

$$\mathbf{Y}_i(t+1) \;=\; W_{ii}(t)\mathbf{Y}_i(t) + \sum_{j \in \mathcal{N}_i(t)} W_{ij}(t)\mathbf{Y}_j(t) \tag{4.33}$$

where $W_{ij}(t) \in \mathbb{R}$ are weight coefficients. Various algorithms adopt various strategies to assign these weight coefficients. For example, (Xiao et al., 2005) suggests to use the so-called Metropolis weights defined as:

$$W_{ij}(t) \;=\; \begin{cases} \dfrac{1}{1 + \max\{d_i(t), d_j(t)\}} & \text{if } \{i,j\} \in \mathcal{E}(t) \\ 1 - \displaystyle\sum_{l \in \mathcal{N}_i(t)} W_{il}(t) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{4.34}$$

where $d_i(t) = |\mathcal{N}_i(t)|$ denotes the degree (number of neighbours) of a node. The Metropolis weights have a notable property that they do not depend on any global knowledge of the network, not even the total number of nodes k. Nevertheless, other weighting schemes have been proposed in the literature with different convergence and topology characteristics, for example, the maximum-degree weighting scheme also described in (Xiao et al., 2005), or a similar scheme

developed in the (Alighanbari and How, 2006)'s extension of the Kalman Consensus algorithm from (Ren et al., 2005).

Note that the composite information vector and matrix do not represent the "real" state estimate per se, but their combination $\mathbf{Y}_i(t)^{-1}\,\hat{\mathbf{y}}_i(t)$ already does, and an algebraic expression of the corresponding error covariance also exists. Assuming that the infinitely occurring communication graphs are jointly connected, it can be proven that the local state estimate converges to the centralised globally-optimal state estimate over time, i.e.

$$\lim_{t\to\infty} \mathbf{Y}_i(t)^{-1}\,\hat{\mathbf{y}}_i(t) \;=\; \left(\sum_{i=1}^{k} \mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{H}_i\right)^{-1} \left(\sum_{i=1}^{k} \mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{z}_i\right) \tag{4.35}$$

The algorithm also provides reasonable intermediate state estimates before the convergence, which are guaranteed to be unbiased (considering the communication graph as a random variable). Unfortunately, this does not imply that the intermediate estimates are consistent and not affected by the common past information problem.

4.5.2 Dynamic Systems

A Kalman Consensus algorithm for dynamical systems can be developed from an algorithm for static systems, such as the one described in the previous section, simply by combining it with the traditional Kalman filter, as shown in (Spanos and Murray, 2005). In such an algorithm, every iteration of the Kalman filter is interlaced with l iterations of the consensus algorithm. For $l = 0$ the algorithm is effectively just a set of local non-communicating Kalman filters, for $l \to \infty$ the algorithm provides globally optimal state estimates, and for any other $l > 0$, the algorithm only provides approximate estimates, with no guarantee on their consistency. Although the quality of the estimates can be partially improved if the Kalman gains and the consensus weights are computed together and not independently of each other, in particular if l is low, as shown in (Carli et al., 2008), such a modification still does not guarantee consistency of the intermediate estimates.

An alternative Kalman Consensus algorithm has been proposed in (Olfati-Saber, 2009), in which the neighbouring nodes exchange not only their state estimates, but also the raw sensor observations from the current time step. The communicated estimates are added to the local estimate with weights derived purely from their covariance. However, although this algorithm is scalable and, under certain assumptions, it is also stable and eventually reaches a consensus, the resulting estimates are only approximate and their consistency is also not guaranteed. The

author of the paper argues that a modification of the algorithm to guarantee the optimality of the estimates requires an $O(k^2)$ computation on every node at every step, making the algorithm not scalable to larger sensor networks.

Another sub-optimal Kalman Consensus algorithm was proposed in (Alriksson and Rantzer, 2006). However, this algorithm has two key weaknesses that make it inapplicable to many practical problems. First, the weights need to be precomputed off-line, which means that the network topology and sensor parameters (covariances) must be constant over time. Second, the nodes have no means to compute covariances of their current estimates.

4.5.3 Discussion

The Kalman Consensus algorithms represent a robust and scalable approach to data fusion in distributed sensor networks. Theoretical properties of these algorithms have been studied extensively in the context of spectral graph theory, with the main result that the speed of convergence of local estimates is proportional to the algebraic connectivity of the sensor network, which is characterised by the second smallest eigenvalue of the Laplacian matrix of the network graph (Olfati-Saber et al., 2007). Also, there is an inherent trade-off between the accuracy of the estimates and the number of iterations of the consensus algorithm. Unfortunately, Kalman Consensus algorithms in general only guarantee consistency of the estimates in the limit of an infinite number of iterations; for intermediate estimates there are typically no such guarantees, which makes an application of these algorithms problematic in the context of safety-critical systems. On the other hand, the new CPI-EnKF and Augmented EnKF update rules developed in this thesis can also be applied in arbitrary sensor networks, while the level of consistency of its estimates at any time only depends on the configurable number of Monte Carlo samples used to represent the estimates, and not the topology of the network as in the case of Kalman Consensus algorithms.

4.6 Covariance Intersection

All of the approaches to the problem of state estimation in the presence of cross-correlations between the errors associated with the *a priori* state estimate and the observation that have been described so far in this thesis operate in the context of the optimal Kalman filtering framework. In other words, these approaches attempt to provide statistically optimal state estimates given all the available sensor observations, which often comes at a price of a limited

applicability due to overly restrictive assumptions, non-scalable computational requirements, or inconsistent intermediate estimates. The *Covariance Intersection* (CI) algorithm, first described in (Uhlmann, 1995) and (Julier and Uhlmann, 1997a), is an elegant approach that addresses such an estimation problem from an entirely different perspective, as it assumes that the cross-correlations are present but unknown, and it provides estimates that are nevertheless consistent and optimal with respect to some other metric, such as the trace or determinant of the estimate's covariance matrix.

In the literature, the CI is typically described as an algorithm operating with estimates of arbitrary random vectors, unrelated to the Kalman filter. However, owing to the topic of this thesis, the CI will be described as an alternative to the standard Kalman filter update rule that is applicable in the case of an unknown cross-correlation between the state estimate error and the observation noise, using the same notation as elsewhere in this chapter. Moreover, the definition of the CI below is based on the information form of the Kalman filter, which leads to a simpler and more comprehensible set of equations.

Definition 4.2 (Covariance Intersection). Suppose a physical system is modelled as a linear discrete-time dynamical system from Definition 2.1. Furthermore, suppose the *a priori* estimate of the system state and an associated Gaussian error are represented using an information vector $\hat{\mathbf{y}}(t) \in \mathbb{R}^n$ and information matrix $\mathbf{Y}(t) \in \mathbb{R}^{n \times n}$, respectively. An observation $\mathbf{z}(t) \in \mathbb{R}^m$ is related to the physical system using an observation model $\mathbf{H} \in \mathbb{R}^{m \times n}$, and it is affected by an observation noise with a covariance $\mathbf{R} \in \mathbb{R}^{m \times m}$. The state estimate error and the observation noise are assumed correlated, but the exact correlation is unknown. In such a system, an *a posteriori* state estimate and an associated error, represented as an information vector $\hat{\mathbf{y}}(t) \in \mathbb{R}^n$ and an information matrix $\mathbf{Y}(t) \in \mathbb{R}^{n \times n}$, respectively, can be computed using the following equations:

$$\hat{\mathbf{y}}(t) = \omega\, \hat{\mathbf{y}}^{-}(t) + (1-\omega)\, \mathbf{H}^{\mathrm{T}}\, \mathbf{R}^{-1}\, \mathbf{z}(t) \tag{4.36}$$

$$\mathbf{Y}(t) = \omega\, \mathbf{Y}^{-}(t) + (1-\omega)\, \mathbf{H}^{\mathrm{T}}\, \mathbf{R}^{-1}\, \mathbf{H} \tag{4.37}$$

where $\omega \in [0, 1]$ is a free parameter.

The name of the algorithm stems from a geometrical interpretation of the CI equations. As illustrated in Figure 4.6, a confidence region associated with the output estimate's covariance always encompasses the intersection of the confidence regions associated with the input estimate's covariances. The weight parameter $\omega \in [0, 1]$ only affects the shape of the confidence region. If

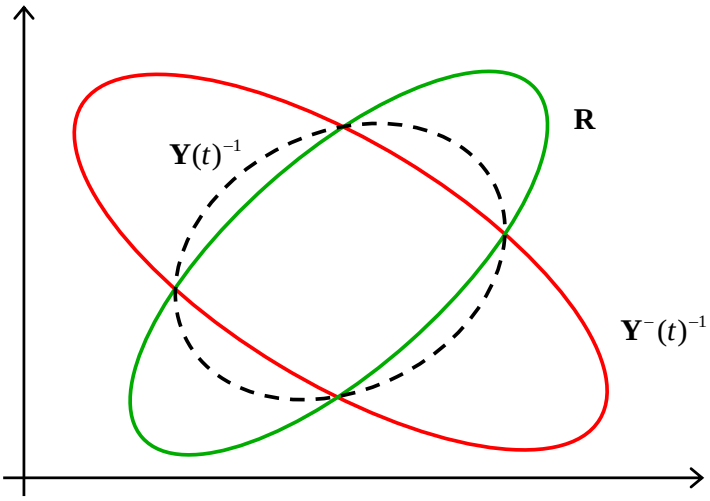


Fig. 4.6: Covariance Intersection algorithm. In this example, $\mathbf{H}$ is an identity matrix.

$\omega = 1$ then the output *a posteriori* estimate equals to the input *a priori* state estimate, and symmetrically, if $\omega = 0$ then it equals to the input observation. The CI algorithm assumes that all the probability distributions involved are Gaussian, however, note that a generalisation to arbitrary probability distributions has been recently proposed in (Bailey et al., 2012).

Although the CI update rule provides a consistent state estimate regardless of the choice of the parameter $\omega \in [0, 1]$, naturally one wants to choose a value that provides an optimal *a posteriori* estimate with respect to some metric on the associated covariance matrix $\mathbf{Y}(t)^{-1}$, typically a value that minimises its trace or determinant. If the cost function is convex with respect to the parameter ω, there is a single distinct optimal value of the parameter. Unfortunately, for determinant and trace, there is no closed-form formula that provides the optimal value ω, and therefore an additional non-linear optimisation step needs to be performed, using an iterative algorithm such as the Newton-Raphson method or the gradient descent. Note that a poor choice of ω can lead to an estimate that is "worse" than any of the input estimates, causing the filter to diverge. The complexity associated with the computation of the optimal value of parameter ω led to development of several fast non-iterative approximations. For example, (Niehsen, 2002) suggests to use the following simple expression for ω:

$$\omega = \frac{\mathrm{tr}(\mathbf{R})}{\mathrm{tr}(\mathbf{Y}^{-}(t)^{-1}) + \mathrm{tr}(\mathbf{R})} \tag{4.38}$$

where $\mathrm{tr}(\mathbf{A})$ denotes the trace of a matrix $\mathbf{A}$. Alternatively, a slightly more involved formula for ω based on information matrices rather than the covariances is provided in (Fränken and Hüpper, 2005).

Similarly to the update rule of the original Information filter described in Equations (2.34)-(2.35), the CI update rule can be easily extended to an arbitrary number k of sensor observations available at a time t as:

$$\hat{\mathbf{y}}(t) \;=\; \omega_0 \, \hat{\mathbf{y}}^{-}(t) \;+ \sum_{i=1}^{k} \omega_i \, \mathbf{H}_i^{\mathrm{T}} \, \mathbf{R}_i^{-1} \, \mathbf{z}_i(t) \tag{4.39}$$

$$\mathbf{Y}(t) \;=\; \omega_0 \, \mathbf{Y}^{-}(t) + \sum_{i=1}^{k} \omega_i \, \mathbf{H}_i^{\mathrm{T}} \, \mathbf{R}_i^{-1} \, \mathbf{H}_i \tag{4.40}$$

where the weight parameters are such as that $\sum_{i=0}^{k} \omega_i = 1$, and the respective observations $\mathbf{z}_i(t) \in \mathbb{R}^{m_i}$ are characterised by an observation model $\mathbf{H}_i \in \mathbb{R}^{m_i \times n}$ and a covariance matrix $\mathbf{R}_i = \mathrm{cov}(\mathbf{z}_i(t)) \in \mathbb{R}^{m_i \times m_i}$.

The CI update rule can be used as a building block for a general and scalable data fusion algorithm for arbitrary sensor networks, as it has been described in (Julier and Uhlmann, 2001a). Assume a switching sensor network $\mathcal{G}(t) = (\mathcal{A},\, \mathcal{E}(t))$ as in Definition 4.1. Each node $\mathcal{A}_i$ models the physical system locally as:

$$\mathbf{x}(t) \;=\; \mathbf{F} \, \mathbf{x}(t-1) + \mathbf{w}(t) \tag{4.41}$$

$$\mathbf{z}_i(t) \;=\; \mathbf{H}_i \, \mathbf{x}(t) + \mathbf{v}_i(t) \tag{4.42}$$

where $\mathbf{z}_i(t) \in \mathbb{R}^{m_i}$ is a local observation, whose relation to the global system state is given by an observation model $\mathbf{H}_i \in \mathbb{R}^{m_i \times n}$, and which is affected by a local observation noise $\mathbf{v}_i(t) \in \mathbb{R}^{m_i}$ with covariance $\mathbf{R}_i = \mathrm{cov}(\mathbf{v}_i(t)) \in \mathbb{R}^{m_i \times m_i}$. The node represents an estimate of the system state at a time t using an information vector $\hat{\mathbf{y}}_i(t) \in \mathbb{R}^n$ and an information matrix $\mathbf{Y}_i(t) \in \mathbb{R}^{n \times n}$. In order to maintain such an estimate, each node recursively performs the following steps:

1. **Prediction**

   Given a local state estimate from the previous time step $t-1$, represented using an information vector $\hat{\mathbf{y}}_i(t-1)$ and an information matrix $\mathbf{Y}_i(t-1)$, compute the *a priori* state estimate, represented as $\hat{\mathbf{y}}_i^{-}(t) \in \mathbb{R}^n$ and $\mathbf{Y}_i^{-}(t) \in \mathbb{R}^{n \times n}$, using the prediction step of the standard Kalman filter:

$$\hat{\mathbf{y}}_i^{-}(t) \;=\; \mathbf{Y}_i^{-}(t) \, \mathbf{F} \, \mathbf{Y}_i^{-1}(t-1) \, \hat{\mathbf{y}}_i(t-1) \tag{4.43}$$

$$\mathbf{Y}_i^{-}(t) \;=\; \left[\mathbf{F} \, \mathbf{Y}_i^{-1}(t-1) \, \mathbf{F}^{\mathrm{T}} \;+\; \mathbf{Q} \right]^{-1} \tag{4.44}$$

2. **Temporary update**

Assimilate a local observation $\mathbf{z}_i(t) \in \mathbb{R}^{m_i}$ into the *a priori* estimate to obtain a *temporary estimate*, represented as an information vector $\tilde{\mathbf{y}}_i(t) \in \mathbb{R}^n$ and information matrix $\tilde{\mathbf{Y}}_i(t) \in \mathbb{R}^{n \times n}$, using the update step equations of the Information filter:

$$\tilde{\mathbf{y}}_i(t) \;=\; \hat{\mathbf{y}}_i^-(t) \;+\; \mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{z}_i(t) \tag{4.45}$$

$$\tilde{\mathbf{Y}}_i(t) \;=\; \mathbf{Y}_i^-(t) \;+\; \mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{H}_i \tag{4.46}$$

3. **Communication**

   Send the temporary estimate $(\tilde{\mathbf{y}}_i(t), \tilde{\mathbf{Y}}_i(t))$ to all neighbouring nodes, and receive their temporary estimates $\{(\tilde{\mathbf{y}}_j(t), \tilde{\mathbf{Y}}_j(t)) \mid j \in \mathcal{N}_i(t)\}$ in return. Recall that $\mathcal{N}_i(t)$ denotes a set of indices of neighbours of a node $\mathcal{A}_i$ at a time t.

4. **Final update**

   Compute the final *a posteriori* state estimate by assimilating the neighbours' temporary estimates (considered as observations) and the local sensor observation $\mathbf{z}_i(t)$ into the *a priori* state estimate. Since the temporary estimates of the neighbours might be correlated to the local *a priori* estimate, they need to be assimilated using the CI update rule, while the sensor observation is affected by a white Gaussian noise and hence it can be assimilated using the update rule of the Information filter. Both these operations can be combined into a single update rule:

$$\hat{\mathbf{y}}_i(t) \;=\; \omega_0\,\hat{\mathbf{y}}_i^-(t) \;+\; \sum_{j \in \mathcal{N}_i(t)} \omega_j\,\tilde{\mathbf{y}}_j(t) \;+\; \mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{z}_i(t) \tag{4.47}$$

$$\mathbf{Y}_i(t) \;=\; \omega_0\,\mathbf{Y}_i^-(t) \;+\; \sum_{j \in \mathcal{N}_i(t)} \omega_j\,\tilde{\mathbf{Y}}_j(t) \;+\; \mathbf{H}_i^{\mathrm{T}}\,\mathbf{R}_i^{-1}\,\mathbf{H}_i \tag{4.48}$$

   where the weight parameters are such that $\sum_{j \in \{0\} \cup \mathcal{N}_i(t)} \omega_j = 1$ and their value is to be optimised as discussed previously.

The CI-based data fusion algorithm described above is guaranteed to provide consistent state estimates at any time. Note that the algorithm is somewhat similar to the Channel filters algorithm described in Section 4.4. The principal difference between the two is that the CI-based algorithm does not make any assumptions on the network topology - the communication links can change over time, the network might contain cycles, and nodes can join and leave it at any time. The algorithm will also work with unidirectional communication links, and all the nodes can operate asynchronously. Additionally, the algorithm can be easily modified so that each node employs a different model of the system.

The CI update rule represents an optimal strategy to update a state estimate using an observation, assuming the correlation between their errors is not known. However, if there is a certain knowledge about the correlation, the CI no longer represents an optimal fusion strategy. Suppose that both the *a priori* state estimate error and the observation error can be split into two components: a known independent error component and a potentially correlated error component. The CI algorithm has no means to take advantage of such a knowledge and therefore it would provide sub-optimal state estimates. This deficiency lead to the development of the so-called *Split Covariance Intersection* (Split CI) algorithm, first published in (Julier and Uhlmann, 2001a) and (Julier and Uhlmann, 2001b), which combines and, in fact, generalises both the CI update rule and the traditional Kalman filter update rule.

Definition 4.3 (Split Covariance Intersection). Suppose a physical system is modelled as a linear discrete-time dynamical system from Definition 2.1. Furthermore, suppose the *a priori* state estimate $\hat{\mathbf{x}}^-(t) \in \mathbb{R}^n$ at a time t has a Gaussian error which can be split into a correlated and independent Gaussian component, with covariances $\mathbf{P}_C^-(t) \in \mathbb{R}^{n\times n}$ and $\mathbf{P}_I^-(t) \in \mathbb{R}^{n\times n}$, respectively. Similarly, the Gaussian noise affecting the observation $\mathbf{z}(t) \in \mathbb{R}^m$ can be split into a correlated and independent Gaussian component with covariances $\mathbf{R}_C \in \mathbb{R}^{m\times m}$ and $\mathbf{R}_I \in \mathbb{R}^{m\times m}$, respectively. The exact correlation between the *a priori* state estimate error and the observation noise is not known. In such a system, an *a posteriori* state estimate $\hat{\mathbf{x}}(t) \in \mathbb{R}^n$ and an associated error covariance $\mathbf{P}(t) \in \mathbb{R}^{n\times n}$ can be computed using the following equations:

$$\hat{\mathbf{x}}(t) = \mathbf{P}(t)\left[\mathbf{A}^{-1}\,\hat{\mathbf{x}}^-(t) + \mathbf{H}^{\mathrm{T}}\,\mathbf{B}^{-1}\,\mathbf{z}(t)\right] \tag{4.49}$$

$$\mathbf{P}(t) = \left[\mathbf{A}^{-1} + \mathbf{H}^{\mathrm{T}}\,\mathbf{B}^{-1}\,\mathbf{H}\right]^{-1} \tag{4.50}$$

$$\mathbf{P}_C(t) = \mathbf{P}(t) - \mathbf{P}_I(t) \tag{4.51}$$

$$\mathbf{P}_I(t) = \mathbf{P}(t)\left[\mathbf{A}^{-1}\,\mathbf{P}_I^-(t)\,\mathbf{A}^{-1} + \mathbf{H}^{\mathrm{T}}\,\mathbf{B}^{-1}\,\mathbf{R}_I\,\mathbf{B}^{-1}\,\mathbf{H}\right]\mathbf{P}(t) \tag{4.52}$$

$$\mathbf{A}^{-1} = \omega\left[\mathbf{P}_C^-(t) + \omega\,\mathbf{P}_I^-(t)\right]^{-1} \tag{4.53}$$

$$\mathbf{B}^{-1} = (1-\omega)\left[\mathbf{R}_C + (1-\omega)\,\mathbf{R}_I\right]^{-1} \tag{4.54}$$

where $\omega \in [0,1]$ is a free parameter. Note that $\mathbf{A}^{-1}$ and $\mathbf{B}^{-1}$ are time-varying matrices, and hence they should rather be denoted $\mathbf{A}^{-1}(t)$ and $\mathbf{B}^{-1}(t)$, respectively, but the shorter notation makes the equations more readable. The resulting estimate's error is also split into a correlated and independent component, with covariances $\mathbf{P}_C(t)$ and $\mathbf{P}_I(t)$, respectively, enabling application of the Split CI algorithm in a recursive filter.

Similarly to the case of the basic CI update rule, the optimal value of the parameter ω needs to be computed using a separate non-linear optimisation algorithm, with respect to some cost function on the resulting estimate. Also, the Split CI can be easily extended to more observations, and it can be used as a building block of a general scalable data fusion algorithm applicable in sensor networks, similar to the one described above. However, the details are beyond the scope of this thesis.

The versatility of the (Split) CI algorithm comes at a price that the estimates are typically overpessimistic when compared to the theoretically optimal estimates. From its nature, the algorithm does not guarantee convergence of the uncertainty associated with the estimates to zero. Also, the additional non-linear optimisation step required to compute correct weight parameters represents an additional technical and computational overhead. The new CPI-EnKF and Augmented EnKF algorithms developed in this thesis aim to provide an equally versatile update rule as the CI algorithm, but one that provides (asymptotically) optimal estimates and ensures convergence.

4.7 Graphical Models

The problem of common past information in sensor networks has also been studied in the context of Bayesian networks (McLaughlin et al., 2004; Brehard and Krishnamurthy, 2007), and from the perspective of information flow in graphs analysed using so-called *graphical models* (Chong and Mori, 2004; McLaughlin et al., 2005; Cetin et al., 2006). Generally speaking, these approaches employ various means to organise the data flow in the network, including stamping and monitoring of the messages, in order to keep track of any dependency and common past information. Some of the approaches also go beyond the Kalman filter as they support non-Gaussian probability distributions. However, these methods are rather application-specific and they do not provide a generally applicable update rule operating in the presence of correlations between state estimate errors and observation errors, which is the primary concern of this thesis, and therefore the methods will not be described in detail.

4.8 Chapter Summary

This chapter discussed the state-of-the-art approaches to linear state estimation in sensor networks. These can be divided into two groups. The first group of approaches, the centralised

Kalman filter and the Decentralised Kalman filter, avoids the common past information problem by communicating raw sensor measurements. Although these methods provide globally-optimal state estimates, they are not scalable to larger sensor networks. On the other hand, the approaches in the second group communicate local state estimates instead of raw sensor measurements, and thus are affected by the common past information problem, which causes correlation between the errors associated with state estimates and observations. Although these approaches can be scaled to arbitrarily large sensor networks, they have other limitations. Channel filters are limited to a tree-connected topology, Kalman Consensus algorithms provide no guarantees of consistency of intermediate local estimates, and the CI is overpessimistic. The new CPI-EnKF and Augmented EnKF update rules developed in this thesis, which will be presented in Chapter 5, facilitate distributed data fusion algorithms that are scalable to arbitrary large sensor networks, provide consistent state estimates and have better accuracy than the CI algorithm.

Chapter 5

Correlations and the Ensemble Kalman Filter

This chapter presents the main contribution of this thesis - generalisation of the Ensemble Kalman Filter (EnKF) to estimation problems characterised by correlated noises or common past information. As will be demonstrated, the prediction step of the EnKF requires only a few modifications in order to support an arbitrary correlation in the process noise, and therefore the problem of generalisation reduces to providing a generalised EnKF update rule that can operate in the presence of correlations between the state estimate errors and the observation errors. In this chapter, three such update rules are derived leading to the following three filters: the Generalised EnKF, the Common Past-Invariant EnKF (CPI-EnKF) and the Augmented EnKF. Each of the rules is based on different presumptions, and their practical applicability to various correlated estimation problems is evaluated in Chapter 6. Note that the new update rules are not exclusive to the EnKF and in theory they are applicable to any other variant of the Kalman filter. However, all of these rules require an estimate of the covariance between the state estimate error and the observation error, which in general cannot be provided analytically and therefore, they are only applicable in the context of the EnKF where the covariance can be estimated from Monte Carlo samples. To the best of the author's knowledge, the work presented here is the only existing attempt to leverage the properties of Monte Carlo samples in the EnKF to address the problem of correlation.

5.1 Correlations in the Prediction Step

Recall Definition 2.6 of the non-linear discrete-time dynamical system model:

$$\mathbf{x}_t \;=\; f(\,\mathbf{x}_{t-1}\,) + \mathbf{w}_t \tag{5.1}$$

$$\mathbf{z}_t \;=\; h(\,\mathbf{x}_t\,) + \mathbf{v}_t \tag{5.2}$$

where $\mathbf{x}_t \in \mathbb{R}^n$ denotes the true system state at a time t, the functions $f\colon \mathbb{R}^n \to \mathbb{R}^n$ and $h\colon \mathbb{R}^n \to \mathbb{R}^m$ represent the process and observation models, respectively, and $\mathbf{w}_t \in \mathbb{R}^n$ and $\mathbf{v}_t \in \mathbb{R}^m$ are random vectors representing the process and observation noises, with covariances $\mathbf{Q} = \mathrm{cov}(\mathbf{w}_t) \in \mathbb{R}^{n\times n}$ and $\mathbf{R} = \mathrm{cov}(\mathbf{v}_t) \in \mathbb{R}^{m\times m}$, respectively. The prediction step of the EnKF, as in Definition 2.10, is performed using the following equations:

$$\mathbf{X}_t^- \;= f(\,\mathbf{X}_{t-1}\,) + \mathbf{W}_t \tag{5.3}$$

$$\mathbf{W}_t \;= [\,\mathbf{w}_t^1,\,\ldots,\,\mathbf{w}_t^N\,] \tag{5.4}$$

where the ensemble $\mathbf{X}_{t-1} \in \mathbb{R}^{n\times N}$ represents the state estimate from the previous time step $t-1$, and the ensemble $\mathbf{X}_t^- \in \mathbb{R}^{n\times N}$ represents the *a priori* state estimate at the time step t. The additive ensemble $\mathbf{W}_t \in \mathbb{R}^{n\times N}$ is composed of random Monte Carlo samples, whose purpose is to reflect the error, introduced by the process noise term $\mathbf{w}_t$ in the true state $\mathbf{x}_t$, to the *a priori* estimate $\mathbf{X}_t^-$ of that state. In the further text, such an ensemble $\mathbf{W}_t$ will be called a *noise compensation ensemble*. In order to ensure the estimate $\mathbf{X}_t^-$ is *correct* (i.e. it is consistent and exact, in the limit of an infinite ensemble), the random samples $\mathbf{w}_t^1,\,\ldots,\,\mathbf{w}_t^N \in \mathbb{R}^n$ composing the ensemble $\mathbf{W}_t$ must be generated using a suitable random number generator in harmony with the statistical properties of the process noise, in particular ensuring that:

$$\mathrm{E}[\mathbf{W}_t] \;\simeq\; 0 \tag{5.5}$$

$$\mathrm{cov}(\mathbf{W}_t) \;\simeq\; \mathbf{Q} \tag{5.6}$$

Assuming that the process noise sequence $\{\mathbf{w}_t\}$ is white and independent, as in Definition 2.3, the random samples $\mathbf{w}_t^1,\,\ldots,\,\mathbf{w}_t^N$ must be generated independently of each other and of any other samples generated previously, which can be expressed as follows:

$$\mathbf{w}_t^i \;\leftarrow\; \mathcal{N}(0, \mathbf{Q}) \tag{5.7}$$

for every $i = 1,\,\ldots,\,N$.

As will be shown in the following text, such a sampling scheme can be extended in order to support correlation in the process noise. In principle, the samples need to be generated in a

way that simulates such a correlation. For example, suppose the process noise sequence $\{\mathbf{w}_t\}$ is sequentiality correlated as discussed in Section 3.2, with the following property:

$$\mathbf{w}_t = \boldsymbol{\Psi}\, \mathbf{w}_{t-1} + \varepsilon_{t-1} \tag{5.8}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{n \times n}$ is an auto-correlation coefficient matrix and the sequence $\{\varepsilon_t\}$ is a white noise process with a time-invariant covariance $\boldsymbol{\Sigma} = \mathrm{cov}(\varepsilon_t) \in \mathbb{R}^{n \times n}$. In order to guarantee the correctness of the *a priori* estimate $\mathbf{X}_t^-$, the random samples $\mathbf{w}_t^1, \ldots, \mathbf{w}_t^N$ need to be generated from the previous random samples $\mathbf{w}_{t-1}^1, \ldots, \mathbf{w}_{t-1}^N$ using the following rule:

$$\mathbf{w}_t^i \leftarrow \boldsymbol{\Psi}\, \mathbf{w}_{t-1}^i + \mathcal{N}(0, \boldsymbol{\Sigma}) \tag{5.9}$$

for every $i = 1, \ldots, N$.

Similarly, if there is a type II correlation between the process noise sequence $\{\mathbf{w}_t\}$ and the observation noise sequence $\{\mathbf{v}_t\}$, as provided in Definition 3.2, with the following property:

$$\mathrm{cov}(\mathbf{w}_i, \mathbf{v}_{j-1}) = \begin{cases} \mathbf{C} & i = j \\ 0 & i \neq j \end{cases} \tag{5.10}$$

then the random samples $\mathbf{w}_t^1, \ldots, \mathbf{w}_t^N$ need to be generated in such a way that the ensemble $\mathbf{W}_t$ is correlated to the previous observation ensemble $\mathbf{Z}_{t-1} = [\mathbf{z}_{t-1}^1, \ldots, \mathbf{z}_{t-1}^N] \in \mathbb{R}^{m \times N}$ (see Section 2.6) as:

$$\mathrm{cov}(\, \mathbf{W}_t,\, \mathbf{Z}_{t-1}\,) \simeq \mathbf{C} \tag{5.11}$$

This can be done by generating the samples $\mathbf{w}_t^1, \ldots, \mathbf{w}_t^N$ and $\mathbf{z}_{t-1}^1, \ldots, \mathbf{z}_{t-1}^N$ together using a single random sampling scheme:

$$\begin{bmatrix} \mathbf{z}_{t-1}^i \\ \mathbf{w}_t^i \end{bmatrix} \leftarrow \mathcal{N}\left(\begin{bmatrix} \mathbf{z}_{t-1} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{R} & \mathbf{C}^{\mathrm{T}} \\ \mathbf{C} & \mathbf{Q} \end{bmatrix} \right) \tag{5.12}$$

for every $i = 1, \ldots, N$.

Note that the prediction step of the EnKF can accommodate virtually any type of correlation in the process noise, including a higher-order sequential correlation or a higher-order cross-correlation with the observation noise. This is only limited by the assumption that the random samples of $\mathbf{W}_t$ can be generated in such a way that they reflect the statistical properties of the process and observation noises. This property stems from the nature of the Monte Carlo method, the fact that each ensemble member is processed independently of other members, and the fact that the order of ensemble members is given and fixed. From a theoretical perspective,

the ensemble prediction is equivalent to solving the Fokker-Planck equation using a Markov Chain Monte Carlo (MCMC) method, which is used to prove its validity (Evensen, 2009).

The idea to apply the Monte Carlo method in order to simulate an evolution of a (sequentially correlated) stochastic process is not novel. However, formalisation of this concept in the context of the EnKF is novel, and it is also a necessary prerequisite for development of other methods presented later in this chapter. Note that the requirement of a fixed order of ensemble members is extremely important. For example, consider a situation where the samples of $\mathbf{W}_t$ were generated correctly according to a statistical model of the noises, but then the order of the samples in the ensemble $\mathbf{W}_t$ was randomly reshuffled. As such, the ensemble $\mathbf{W}_t$ would effectively lose information about its correlation to the ensemble $\mathbf{X}_{t-1}$, and the EnKF prediction step would produce an inconsistent estimate. In other words,

The correlation information is encoded in the order of ensemble members.

It is important to note that the EnKF prediction step with a correlated process noise compensation ensemble discussed in this section provides optimal *a priori* estimates only if the filter performs no update step. In order to achieve optimality in a standard filter, the recursive noise compensation ensemble needs to be subject to the update operation of the filter. In other words, if the noise compensation ensemble $\mathbf{W}_t$ is generated recursively from the previous ensemble $\mathbf{W}_{t-1}$, for example, using the rule from Equation (5.9), then $\mathbf{W}_t$ needs to be included in the system state ensemble, so that it is also updated by the filter, leading to an updated noise compensation ensemble $\mathbf{W}_t^+$. In the subsequent prediction step, the ensemble $\mathbf{W}_{t+1}$ must be generated from such an updated ensemble $\mathbf{W}_t^+$ instead of the original noise compensation ensemble $\mathbf{W}_t$. Note that this is formally equivalent to the state augmentation procedure described in Section 3.2. There is, however, an important difference that in the context of the EnKF, the *a priori* estimate will be consistent even if the system state was not augmented with the noise compensation ensemble. This feature is advantageous for applications with high-dimensional state spaces, which cannot afford the performance penalty associated with the state augmentation procedure. Note that a numerical analysis of the points discussed in this paragraph is available in Section 6.3.1.

Although the treatment of correlations in the prediction step of the EnKF using the framework outlined in this section is very straightforward, the same treatment of correlations is not exactly possible in the case of the update step of the EnKF. In the update step, the ensemble members are not processed independently of each other, because the Kalman gain applied to

each of the ensemble members depends on the collective property of all the ensemble members - their covariance. Therefore, the problem of correlation in the update step requires more involved methods, which will be described in the following sections.

5.2 Correlations in the Update Step

Recall that the update step of the EnKF, as given in Definition 2.10, is performed using the following equations:

$$\mathbf{X}_t \;=\; \mathbf{X}_t^- + \mathbf{K}_t\big(\,\mathbf{Z}_t - h(\mathbf{X}_t^-)\,\big) \tag{5.13}$$

$$\mathbf{K}_t \;=\; \mathrm{cov}\big(\mathbf{X}_t^-, h(\mathbf{X}_t^-)\big)\,[\,\mathrm{cov}(h(\mathbf{X}_t^-)) + \mathbf{R}\,]^{-1} \tag{5.14}$$

For brevity, however, this chapter will adopt a slightly different notation of the update rule. Firstly, the time indices in the observation ensemble $\mathbf{Z}_t$ and the Kalman gain $\mathbf{K}_t$ will be dropped, the *a priori* state estimate ensemble $\mathbf{X}_t^-$ denoted simply as $\mathbf{X}$, and the *a posteriori* state estimate ensemble $\mathbf{X}_t$ as $\mathbf{X}^+$. Secondly, the observation model h will be assumed linear, enabling a substitution $\mathbf{HX} = h(\mathbf{X})$. Note that this is a standard assumption in all the EnKF-related proofs, as discussed in Section 2.6. And finally, the observation ensemble $\mathbf{Z}_t$ does not necessarily need to be generated by a random sampling scheme from a single observation vector $\mathbf{z}_t$, leading to a substitution $\mathrm{cov}(\mathbf{Z}) \simeq \mathbf{R}$. The last modification makes the update rule applicable in situations where the observations are already represented using ensembles, such as in sensor networks in which the communicated state estimates are treated as observations. All these modifications combined result in the following simplified EnKF update rule:

$$\mathbf{X}^+ \;=\; \mathbf{X} + \mathbf{K}(\,\mathbf{Z} - \mathbf{HX}\,) \tag{5.15}$$

$$\mathbf{K} \;=\; \mathrm{cov}(\mathbf{X}, \mathbf{HX})\;[\,\mathrm{cov}(\mathbf{HX}) + \mathrm{cov}(\mathbf{Z})\,]^{-1} \tag{5.16}$$

where

$\mathbf{X}$	$\in \mathbb{R}^{n\times N}$	(*a priori* state estimate ensemble)
$\mathbf{X}^+$	$\in \mathbb{R}^{n\times N}$	(*a posteriori* state estimate ensemble)
$\mathbf{Z}$	$\in \mathbb{R}^{m\times N}$	(observation ensemble)
$\mathbf{H}$	$\in \mathbb{R}^{m\times n}$	(observation model)
$\mathbf{K}$	$\in \mathbb{R}^{n\times m}$	(Kalman gain factor)

From a theoretical perspective, the new update rule is equivalent to the original EnKF update rule, as it ensures optimality of the *a posteriori* state estimate, in the limit of an infinite ensemble

and given all the standard assumptions, as discussed in Section 2.6. In particular, the assumption that the process noise sequence $\{\mathbf{w}_t\}$ and the observation noise sequence $\{\mathbf{v}_t\}$ are both white and independent, as in Definition 2.3, implies the following assumption:

$$\operatorname{cov}(\,\mathbf{X},\,\mathbf{Z}\,) \simeq 0 \tag{5.17}$$

However, if the observation ensemble $\mathbf{Z}$ is generated from a single physical observation $\mathbf{z}_t \in \mathbb{R}^m$ using a random sampling scheme, as described in Section 2.6, and there is some form of correlation in the observation noise sequence $\{\mathbf{v}_t\}$, such as a first-order sequential correlation discussed in Section 3.3, or the type I cross-correlation between the process and observation noise sequences discussed in Section 3.1, the random samples of $\mathbf{Z}$ must be generated in a way that simulates such correlations, in very much the same way as in the case of correlations in the process noise as discussed in Section 5.1. Alternatively, the observation ensemble $\mathbf{Z}$ might be derived from a state estimate ensemble maintained by another node in a sensor network, and therefore the observation ensemble $\mathbf{Z}$ and the local state estimate $\mathbf{X}$ might share common past information, as discussed in Chapter 4. In effect, all these factors can potentially lead to a violation of the assumption in Equation (5.17), formally:

$$\operatorname{cov}(\,\mathbf{X},\,\mathbf{Z}\,) \not\simeq 0 \tag{5.18}$$

The following sections analyse various modifications of the EnKF update rule with the aim to provide a consistent estimate even if this eventuality occurs.

5.2.1 Generalised Ensemble Kalman Filter

The most natural candidate for a modification of the EnKF update rule to support a correlation between the state estimate error and observation error is based on the Generalised Kalman filter, introduced in Definition 3.1, which supports the type I correlation between the process and observation noises. In this type of correlation, the process noise sequence $\{\mathbf{w}_t\}$ and the observation noise sequence $\{\mathbf{v}_t\}$ are assumed to be mutually correlated with the following property:

$$\operatorname{cov}(\mathbf{w}_i, \mathbf{v}_j) = \begin{cases} \mathbf{C} & i = j \\ 0 & i \neq j \end{cases} \tag{5.19}$$

where $\mathbf{C} \in \mathbb{R}^{n \times m}$ represents a cross-covariance matrix. The Generalised Kalman filter defines an optimal update rule for this eventuality, which extends the standard Kalman filter update rule with several terms involving the cross-covariance matrix $\mathbf{C}$. In fact, the formula for the a

posteriori mean state estimate from Equation (3.4) only differs from the standard formula from Equation (2.16) in the Kalman gain factor, which is newly defined by Equation (3.6).

In the EnKF, if there is a correlation between the ensembles $\mathbf{X}$ and $\mathbf{Z}$, it can be assumed that it is caused by a correlation between the latest process and observation noise terms, whose magnitude $\mathbf{C}$ can be estimated as:

$$\mathbf{C} \simeq \mathrm{cov}(\mathbf{X}, \mathbf{Z}) \tag{5.20}$$

Consequently, by noting that the EnKF update rule performs with each pair of ensemble members from the state estimate and the observation practically the same computation as the traditional Kalman filter update rule performs with the mean state estimate and the observation vector, one can speculate that a simple replacement of the standard Kalman gain Equation (5.16) in the EnKF update rule by the generalised Kalman gain Equation (3.6), in this context defined as:

$$\mathbf{K} = [\,\mathrm{cov}(\mathbf{X}, \mathbf{HX}) + \mathrm{cov}(\mathbf{X}, \mathbf{Z})\,]\,[\,\mathrm{cov}(\mathbf{HX}) + \mathrm{cov}(\mathbf{HX}, \mathbf{Z}) + \mathrm{cov}(\mathbf{Z}, \mathbf{HX}) + \mathrm{cov}(\mathbf{Z})\,]^{-1} \tag{5.21}$$

will lead to a generalised EnKF update rule that would provide optimal state estimates even in the presence of correlations. Formally, the complete update rule of such a hypothetical *Generalised EnKF* is defined as follows:

$$\mathbf{X}^{+} = \mathbf{X} + \mathbf{K}(\,\mathbf{Z} - \mathbf{HX}\,) \tag{5.22}$$

$$\mathbf{K} = \mathrm{cov}(\mathbf{X}, \mathbf{HX} + \mathbf{Z})\ \mathrm{cov}(\mathbf{HX} + \mathbf{Z})^{-1} \tag{5.23}$$

Note that the simplification of Equation (5.21) into Equation (5.23) is due to the fact that covariance is a bilinear operator. However, as it will be demonstrated using numerical simulations in Section 6.3.2, such an update rule does **not** provide an optimal *a posteriori* state estimate. This finding represents one of the contributions of this thesis.

5.2.2 Common Past-Invariant Ensemble Kalman Filter

This section introduces another variant of the EnKF update rule, which operates in the presence of correlations between the state estimate and observation errors. This new update rule, however, is based on a different interpretation of the origin of correlation than the Generalised EnKF method described in the previous section, as it assumes that the correlation is caused exclusively by the presence of a zero-mean additive error term shared between the state estimate error and the observation errors. The goal of the new update rule is to compute an *a posteriori* state

estimate equivalent to a situation where the shared error term was not present. The resulting modification of the EnKF is known as the *Common Past-Invariant Ensemble Kalman Filter* (CPI-EnKF), and it has been first published in (Čurn et al., 2012b).

Assume an *a priori* state estimate ensemble $\mathbf{X} \in \mathbb{R}^{n \times N}$ and an observation ensemble $\mathbf{Z} \in \mathbb{R}^{m \times N}$. The zero-mean Gaussian errors associated with the state estimate and the observation are fully represented by ensembles $\mathbf{X} - \mathrm{E}[\mathbf{X}]$ and $\mathbf{Z} - \mathrm{E}[\mathbf{Z}]$, respectively, where $\mathbf{A} - \mathbf{b}$ denotes a column-wise subtraction of a vector $\mathbf{b}$ from each column of an ensemble $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]$, formally $\mathbf{A} - \mathbf{b} = [\mathbf{a}_1 - \mathbf{b}, \ldots, \mathbf{a}_N - \mathbf{b}]$. Similarly, $\mathbf{A} + \mathbf{b} = [\mathbf{a}_1 + \mathbf{b}, \ldots, \mathbf{a}_N + \mathbf{b}]$ denotes a column-wise addition. Suppose that each of the errors is a combination of two zero-mean multivariate Gaussian error terms: a shared error term and two independent error terms. The shared error term can be represented by a hypothetical ensemble $\mathbf{\Sigma} \in \mathbb{R}^{n \times N}$ and the two independent error terms by hypothetical ensembles $\Delta^{\mathrm{X}} \in \mathbb{R}^{n \times N}$ and $\Delta^{\mathrm{Z}} \in \mathbb{R}^{m \times N}$, such as:

$$\mathbf{X} - \mathrm{E}[\mathbf{X}] \;=\; \mathbf{\Sigma} + \Delta^{\mathrm{X}} \tag{5.24}$$

$$\mathbf{Z} - \mathrm{E}[\mathbf{Z}] \;=\; \mathbf{H}\mathbf{\Sigma} + \Delta^{\mathrm{Z}} \tag{5.25}$$

where $\mathbf{H} \in \mathbb{R}^{m \times n}$ is the observation model. The assumptions on the error terms imply the following properties:

$$\mathrm{E}(\mathbf{\Sigma}) \simeq \mathrm{E}(\Delta^{\mathrm{X}}) \simeq \mathrm{E}(\Delta^{\mathrm{Z}}) \simeq \mathrm{cov}(\mathbf{\Sigma}, \Delta^{\mathrm{X}}) \simeq \mathrm{cov}(\Delta^{\mathrm{X}}, \Delta^{\mathrm{Z}}) \simeq \mathrm{cov}(\Delta^{\mathrm{Z}}, \mathbf{\Sigma}) \simeq 0 \tag{5.26}$$

If one could eliminate the shared error from both the state estimate and observation ensembles, i.e. to compute $\mathbf{X} - \mathbf{\Sigma}$ and $\mathbf{Z} - \mathbf{H}\mathbf{\Sigma}$, respectively, and then perform the traditional EnKF update on these *modified* ensembles, the resulting *a posteriori* estimate would effectively ignore the shared error between the state and the observation. Such an update would be optimal, because all the standard EnKF assumptions are satisfied. Unfortunately, it is not possible to evaluate $\mathbf{\Sigma}$ exactly, but as it will be shown in this section, this is not necessary, because an equivalent *a posteriori* estimate can be computed using other means. The goal is to derive a modified EnKF update rule:

$$\mathbf{X}^{+} \;=\; \mathbf{X} + \mathbf{K}\big(\,\mathbf{Z} - \mathbf{H}\mathbf{\Sigma} \;-\; \mathbf{H}[\mathbf{X} - \mathbf{\Sigma}]\,\big) \tag{5.27}$$

$$\mathbf{K} \;=\; \mathrm{cov}(\mathbf{X} - \mathbf{\Sigma}, \mathbf{H}[\mathbf{X} - \mathbf{\Sigma}])\,\big[\,\mathrm{cov}(\mathbf{H}[\mathbf{X} - \mathbf{\Sigma}]) + \mathrm{cov}(\mathbf{Z} - \mathbf{H}\mathbf{\Sigma})\,\big]^{-1} \tag{5.28}$$

which can be rewritten using Equations (5.24) and (5.25) to the following form:

$$\mathbf{X}^{+} \;=\; \mathbf{X} + \mathbf{K}\big(\,\mathbf{Z} - \mathbf{H}\mathbf{X}\,\big) \tag{5.29}$$

$$\mathbf{K} \;=\; \mathrm{cov}(\Delta^{\mathrm{X}}, \mathbf{H}\Delta^{\mathrm{X}})\,[\,\mathrm{cov}(\mathbf{H}\Delta^{\mathrm{X}}) + \mathrm{cov}(\Delta^{\mathrm{Z}})\,]^{-1} \tag{5.30}$$

Note that Equation (5.29) is equivalent to the traditional EnKF update rule as in Equation (5.15), and therefore the issue discussed in this section reduces to a derivation of the new Kalman gain in Equation (5.30).

Due to the assumptions that ensembles $\mathbf{\Sigma}$, Δ^{X} and Δ^{X} represent mutually independent errors, expressed in Equation (5.26), it directly follows from Equations (5.24) and (5.25) that:

$$\operatorname{cov}(\mathbf{Z}-\mathbf{HX}) \simeq \operatorname{cov}(\mathbf{H}\Delta^{\mathrm{X}}) + \operatorname{cov}(\Delta^{\mathrm{Z}}) \tag{5.31}$$

$$\operatorname{cov}(\mathbf{HX}) \simeq \operatorname{cov}(\mathbf{H\Sigma}) + \operatorname{cov}(\mathbf{H}\Delta^{\mathrm{X}}) \tag{5.32}$$

$$\operatorname{cov}(\mathbf{Z}) \simeq \operatorname{cov}(\mathbf{H\Sigma}) + \operatorname{cov}(\Delta^{\mathrm{Z}}) \tag{5.33}$$

Note that such $\operatorname{cov}(\mathbf{Z}-\mathbf{HX})$ from Equation (5.31) asymptotically equals the inverse component of the modified Kalman gain from Equation (5.30), thus only an expression for the component $\operatorname{cov}(\Delta^{\mathrm{X}}, \mathbf{H}\Delta^{\mathrm{X}})$ needs to be derived. Solving the system of asymptotic Equations (5.31)-(5.33) and decomposing $\operatorname{cov}(\mathbf{Z}-\mathbf{HX})$ as:

$$\operatorname{cov}(\mathbf{Z}-\mathbf{HX}) = \operatorname{cov}(\mathbf{Z}) - \operatorname{cov}(\mathbf{Z},\mathbf{HX}) - \operatorname{cov}(\mathbf{HX},\mathbf{Z}) + \operatorname{cov}(\mathbf{HX}) \tag{5.34}$$

leads to the following expression:

$$\operatorname{cov}(\mathbf{H}\Delta^{\mathrm{X}}) \simeq \operatorname{cov}(\mathbf{HX}) - \tfrac{1}{2}\big[\operatorname{cov}(\mathbf{HX},\mathbf{Z}) + \operatorname{cov}(\mathbf{Z},\mathbf{HX})\big] \tag{5.35}$$

Suppose that $m = n$ and $\mathbf{H}$ is invertible. A left-multiplication of Equation (5.35) by $\mathbf{H}^{-1}$ yields the desired missing component $\operatorname{cov}(\Delta^{\mathrm{X}}, \mathbf{H}\Delta^{\mathrm{X}})$ of the modified Kalman gain from Equation (5.30) as:

$$\operatorname{cov}(\Delta^{\mathrm{X}}, \mathbf{H}\Delta^{\mathrm{X}}) \simeq \operatorname{cov}(\mathbf{X},\mathbf{HX}) - \tfrac{1}{2}\big[\operatorname{cov}(\mathbf{X},\mathbf{Z}) + \mathbf{H}^{-1}\operatorname{cov}(\mathbf{Z},\mathbf{HX})\big] \tag{5.36}$$

However, as discussed in Section 2.6, one of the principal advantages of the EnKF is that it does not require the observation model in a matrix form $\mathbf{H}$. This advantage would be lost if $\mathbf{H}^{-1}$ had to be a priori known. Fortunately, by noting that

$$\operatorname{cov}(\mathbf{X},\mathbf{HX}) \simeq \operatorname{cov}(\mathbf{H}^{-1}\mathbf{HX},\mathbf{HX}) \tag{5.37}$$

$$\simeq \mathbf{H}^{-1}\operatorname{cov}(\mathbf{HX}) \tag{5.38}$$

it is possible to approximate $\mathbf{H}^{-1}$ as:

$$\mathbf{H}^{-1} \simeq \operatorname{cov}(\mathbf{X},\mathbf{HX})\operatorname{cov}(\mathbf{HX})^{-1} \tag{5.39}$$

under the assumption that $\mathrm{cov}(\mathbf{HX})$ is invertible. If it is not invertible, the solution for $\mathbf{H}^{-1}$ still exists but it is not unique; more precisely, the solutions for $\mathbf{H}^{-1}$ form a vector space of dimension $m - r$, where $r = \mathrm{rank}(\mathrm{cov}(\mathbf{HX}))$. In this case, an acceptable solution can be found if one considers a reduced observation model $\mathbf{H}_* \in \mathbb{R}^{r \times n}$, which is a matrix obtained from $\mathbf{H}$ by removing $m - r$ rows, such as that $\mathrm{cov}(\mathbf{H}_*\mathbf{X})$ is invertible. This leads to the reduced solution:

$$\mathbf{H}_*^{-\mathbf{1}} \simeq \mathrm{cov}(\mathbf{X}, \mathbf{H}_*\mathbf{X})\,\mathrm{cov}(\mathbf{H}_*\mathbf{X})^{-1} \tag{5.40}$$

which can be extended to the full solution $\mathbf{H}^{-1}$ by simply inserting r zero columns at the positions corresponding to the rows removed from $\mathbf{H}$.

Finally, Equations (5.36) and (5.39) and can be combined into a single expression:

$$\mathrm{cov}(\Delta^{\mathrm{X}}, \mathbf{H}\Delta^{\mathrm{X}}) \simeq \mathrm{cov}(\mathbf{X}, \mathbf{HX}) - \tfrac{1}{2}\,\mathrm{cov}(\mathbf{X}, \mathbf{HX})\,\mathrm{cov}(\mathbf{HX})^{-1}\big[\mathrm{cov}(\mathbf{HX}, \mathbf{Z}) + \mathrm{cov}(\mathbf{Z}, \mathbf{HX})\big] \tag{5.41}$$

Simply speaking, Equation (5.39) says that each row of $\mathrm{cov}(\mathbf{X}, \mathbf{HX})$ can be expressed as a linear combination of rows from $\mathrm{cov}(\mathbf{HX})$, and Equation (5.41) says that each row of $\mathrm{cov}(\Delta^{\mathrm{X}}, \mathbf{H}\Delta^{\mathrm{X}})$ can be expressed as a linear combination of rows from $\mathrm{cov}(\mathbf{H}\Delta^{\mathrm{X}})$, with the same coefficients. Also note that if Equation (5.41) is left-multiplied by $\mathbf{H}$, it collapses to Equation (5.35), and also it can be easily rewritten to the form of Equation (5.36), exactly as expected.

The complete formula of the CPI-EnKF update rule is as follows:

$$\mathbf{X}^+ = \mathbf{X} + \mathbf{K}\big(\,\mathbf{Z} - \mathbf{HX}\,\big) \tag{5.42}$$

$$\mathbf{K} = \Big(\mathrm{cov}(\mathbf{X}, \mathbf{HX}) - \tfrac{1}{2}\,\mathrm{cov}(\mathbf{X}, \mathbf{HX})\,\mathrm{cov}(\mathbf{HX})^{-1}\big[\mathrm{cov}(\mathbf{HX}, \mathbf{Z}) + \mathrm{cov}(\mathbf{Z}, \mathbf{HX})\big]\Big)\,\mathrm{cov}(\mathbf{Z} - \mathbf{HX})^{-1} \tag{5.43}$$

It can be easily seen that if $\mathbf{X}$ and $\mathbf{Z}$ are independent and hence uncorrelated, then the CPI-EnKF update rule reduces to the traditional EnKF update rule as in Equations (5.15)-(5.16) in the limit of an infinite ensemble, because then $\mathrm{cov}(\mathbf{HX}, \mathbf{Z}) \simeq 0$ and from Equation (5.34) it follows that:

$$\mathrm{cov}(\mathbf{Z} - \mathbf{HX})^{-1} \simeq \big[\,\mathrm{cov}(\mathbf{HX}) + \mathrm{cov}(\mathbf{Z})\,\big]^{-1} \tag{5.44}$$

As such, the CPI-EnKF is effectively a generalisation of the EnKF. However, note that the term $\mathrm{cov}(\mathbf{Z} - \mathbf{HX})$ estimates the true covariance less precisely than the original term $\mathrm{cov}(\mathbf{HX}) + \mathrm{cov}(\mathbf{Z})$, given the same number of ensemble members; they are equivalent only in the limit of an infinite ensemble. Another issue is that if $\mathbf{Z}$ and $\mathbf{HX}$ are too "close", the inversion operation $\mathrm{cov}(\mathbf{Z} - \mathbf{HX})^{-1}$ will become ill-conditioned. In order to avoid this problem, a practical implementation of the algorithm should first compute the effective rank of the matrix $\mathrm{cov}(\mathbf{Z} -$

HX), for example, using the singular value decomposition (SVD) method, and skip the update altogether if the rank is not equal to m.

As discussed above, the CPI-EnKF update rule provides an optimal state estimate under the assumption that a correlation between the state estimate and observation errors is caused exclusively by the presence of a shared additive error term. Such an assumption is stricter than the assumption of the Split CI algorithm that both the state estimate and observation error covariances can be split into a known-correlated and independent components, as described in Section 4.6; here the known-correlated components of both the error covariances are assumed equal. Although such an assumption is rarely perfectly satisfied in practice, it will be demonstrated in Chapter 6 that the CPI-EnKF provides reasonable and consistent estimates in many practical correlated estimation problems, in particular in sensor networks where the correlation is caused by common past information shared between the nodes. The most significant feature of the CPI-EnKF, which also gave the filter its name, is the fact that if an observation offers no new information that is not already available in the state estimate, the resulting *a posteriori* estimate is invariant with respect to such an observation. A practical utility of this feature is demonstrated in Section 7.2 in the context of the simultaneous localisation and mapping problem (SLAM) in robotics.

5.2.3 Augmented Ensemble Kalman Filter

This section provides a derivation of an alternative generalisation of the EnKF update rule to support correlation between the state estimate and observation ensembles, using an approach similar to the state augmentation procedure described in Section 3.3.1. As discussed in Chapter 4, nodes in sensor networks may communicate their local state estimates, which other nodes consider as observations. Looking at this from the opposite perspective, an observation can be considered as part of the system state, i.e. a state vector $\mathbf{x}_t \in \mathbb{R}^n$ can be augmented with an observation $\mathbf{z}_t \in \mathbb{R}^m$. Define an augmented state vector $\mathbf{x}_t^* \in \mathbb{R}^{n+m}$ as:

$$\mathbf{x}_t^* = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{z}_t \end{bmatrix} \tag{5.45}$$

and a pseudo-measurement vector $\mathbf{z}_t^* \in \mathbb{R}^m$ as:

$$\mathbf{z}_t^* = 0 \tag{5.46}$$

The pseudo-measurement is related to the augmented state using an augmented observation model $h^*\colon \mathbb{R}^{n+m} \to \mathbb{R}^m$ defined as:

$$h^*(\mathbf{x}, \mathbf{z}) \;=\; h(\mathbf{x}) - \mathbf{z} \tag{5.47}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^m$ are function parameters and $h\colon \mathbb{R}^n \to \mathbb{R}^m$ represents the original observation model. A full observation equation of such an augmented dynamical system model is:

$$\mathbf{z}_t^* \;=\; h^*(\mathbf{x}_t^*) + \mathbf{v}_t \tag{5.48}$$

Note that a corresponding time-evolution equation of the augmented system could also be defined, but that is not necessary, because this section only deals with the update rule.

In the context of the EnKF, with the simplified notation introduced in the beginning of Section 5.2, an *a priori* estimate of the augmented system state is represented using an augmented ensemble $\mathbf{X}^* \in \mathbb{R}^{(n+m)\times N}$ defined as:

$$\mathbf{X}^* \;=\; \begin{bmatrix} \mathbf{X} \\ \mathbf{Z} \end{bmatrix} \tag{5.49}$$

where $\mathbf{X} \in \mathbb{R}^{n\times N}$ denotes an ensemble representing the *a priori* state estimate and $\mathbf{Z} \in \mathbb{R}^{m\times N}$ is the original observation ensemble. The pseudo-measurement ensemble $\mathbf{Z}^* \in \mathbb{R}^{m\times N}$ is a zero matrix, i.e.

$$\mathbf{Z}^* \;=\; 0 \tag{5.50}$$

If the original observation model h is linear, i.e. $h(\mathbf{X}) = \mathbf{H}\mathbf{X}$ for some $\mathbf{H} \in \mathbb{R}^{m\times n}$, then the augmented observation model h^* is also linear, because $h^*(\mathbf{X}^*) = \mathbf{H}^*\mathbf{X}^*$ for a matrix $\mathbf{H}^* \in \mathbb{R}^{m\times(n+m)}$ defined as:

$$\mathbf{H}^* \;=\; [\; \mathbf{H} \quad -I \;] \tag{5.51}$$

From Equation (5.50) it follows that:

$$\mathrm{cov}(\mathbf{X}^*, \mathbf{Z}^*) \;=\; 0 \tag{5.52}$$

and therefore one can speculate that an *a posteriori* estimate of the augmented system state, represented using an ensemble $\mathbf{X}^{*+} \in \mathbb{R}^{(n+m)\times N}$, can be computed using the traditional EnKF update rule:

$$\mathbf{X}^{*+} \;=\; \mathbf{X}^* + \mathbf{K}^*(\,\mathbf{Z}^* - \mathbf{H}^*\mathbf{X}^*\,) \tag{5.53}$$

$$\mathbf{K}^* \;=\; \mathrm{cov}(\mathbf{X}^*, \mathbf{H}^*\mathbf{X}^*)\;[\,\mathrm{cov}(\mathbf{H}^*\mathbf{X}^*) + \mathrm{cov}(\mathbf{Z}^*)\,]^{-1} \tag{5.54}$$

Equation (5.53) can be rewritten as:

$$\mathbf{X}^{*+} = \begin{bmatrix} \mathbf{X} \\ \mathbf{Z} \end{bmatrix} + \mathbf{K}^* \left(0 - [\, \mathbf{H} \;\; -I \,] \begin{bmatrix} \mathbf{X} \\ \mathbf{Z} \end{bmatrix} \right) \tag{5.55}$$

$$= \begin{bmatrix} \mathbf{X} \\ \mathbf{Z} \end{bmatrix} + \mathbf{K}^* \, (\, \mathbf{Z} - \mathbf{H}\mathbf{X} \,) \tag{5.56}$$

and similarly, Equation (5.54) can be rewritten as:

$$\mathbf{K}^* = \mathrm{cov}(\mathbf{X}^*, \mathbf{X}^*) \, \mathbf{H}^{*\mathrm{T}} \left[\mathbf{H}^* \, \mathrm{cov}(\mathbf{X}^*) \, \mathbf{H}^{*\mathrm{T}} + 0 \right]^{-1} \tag{5.57}$$

$$= \begin{bmatrix} \mathrm{cov}(\mathbf{X}) & \mathrm{cov}(\mathbf{X}, \mathbf{Z}) \\ \mathrm{cov}(\mathbf{Z}, \mathbf{X}) & \mathrm{cov}(\mathbf{Z}) \end{bmatrix} \begin{bmatrix} \mathbf{H}^{\mathrm{T}} \\ -I \end{bmatrix} \left([\, \mathbf{H} \;\; -I \,] \begin{bmatrix} \mathrm{cov}(\mathbf{X}) & \mathrm{cov}(\mathbf{X}, \mathbf{Z}) \\ \mathrm{cov}(\mathbf{Z}, \mathbf{X}) & \mathrm{cov}(\mathbf{Z}) \end{bmatrix} \begin{bmatrix} \mathbf{H}^{\mathrm{T}} \\ -I \end{bmatrix} \right)^{-1} \tag{5.58}$$

$$= \begin{bmatrix} \mathrm{cov}(\mathbf{X}, \mathbf{H}\mathbf{X}) - \mathrm{cov}(\mathbf{X}, \mathbf{Z}) \\ \mathrm{cov}(\mathbf{Z}, \mathbf{H}\mathbf{X}) - \mathrm{cov}(\mathbf{Z}) \end{bmatrix} [\, \mathrm{cov}(\mathbf{H}\mathbf{X}) - \mathrm{cov}(\mathbf{H}\mathbf{X}, \mathbf{Z}) - \mathrm{cov}(\mathbf{Z}, \mathbf{H}\mathbf{X}) + \mathrm{cov}(\mathbf{Z}) \,]^{-1} \tag{5.59}$$

$$= \begin{bmatrix} \mathrm{cov}(\mathbf{X}, \mathbf{H}\mathbf{X} - \mathbf{Z}) \\ \mathrm{cov}(\mathbf{Z}, \mathbf{H}\mathbf{X} - \mathbf{Z}) \end{bmatrix} \mathrm{cov}(\mathbf{H}\mathbf{X} - \mathbf{Z})^{-1} \tag{5.60}$$

Considering that the first n rows of the *a posteriori* estimate $\mathbf{X}^{*+}$ represent the *a posteriori* estimate of the original (non-augmented) system state, denoted as an ensemble $\mathbf{X}^+ \in \mathbb{R}^{n \times N}$, it is possible to extract the full *Augmented EnKF* update rule from Equations (5.56) and (5.60) as follows:

$$\mathbf{X}^+ = \mathbf{X} + \mathbf{K}(\, \mathbf{Z} - \mathbf{H}\mathbf{X} \,) \tag{5.61}$$

$$\mathbf{K} = \mathrm{cov}(\mathbf{X}, \mathbf{H}\mathbf{X} - \mathbf{Z}) \, \mathrm{cov}(\mathbf{H}\mathbf{X} - \mathbf{Z})^{-1} \tag{5.62}$$

Interestingly, the Augmented EnKF update rule differs from the Generalised EnKF update rule, presented in the previous section, only in the sign of the term $\mathbf{Z}$, and it also constitutes a generalisation of the original EnKF in the limit of an infinite ensemble: if $\mathrm{cov}(\mathbf{X}, \mathbf{Z}) = 0$ then Equations (5.61) and (5.62) reduce to the traditional EnKF update rule, as provided in Equations (5.15) and (5.16), respectively. Note that for one-dimensional state spaces, the Augmented EnKF update rule is algebraically equivalent to the CPI-EnKF update rule derived in Section 5.2.2. Nevertheless, as will be demonstrated in Chapter 6 using numerical simulations, the Augmented EnKF provides de facto equivalent results as the CPI-EnKF even in higher-dimensional problems, while exhibiting a better numerical stability. Also, similarly to the case

of the CPI-EnKF, special care needs to be taken to avoid the potentially ill-conditioned inversion $\mathrm{cov}(\mathbf{HX} - \mathbf{Z})^{-1}$, as discussed in Section 5.2.2.

5.2.4 Discussion

All the variants of the EnKF update rule presented in this chapter are based on the traditional assumptions that all the probability distributions involved are Gaussian, the dynamical system is linear and that it models the physical system exactly. However, similarly to the case of any other variant of the Kalman filter, the new update rules might be applied in practical estimation problems where the formal assumptions are not exactly satisfied, while providing reasonable state estimates. As per the assumption of linearity of the observation model, all the newly developed update rules only use the observation model in the form of term $\mathbf{HX}$. Therefore, if the observation model is non-linear and expressed using a function $h\colon \mathbb{R}^n \to \mathbb{R}^m$ instead of a matrix $\mathbf{H} \in \mathbb{R}^{m \times n}$, the term $\mathbf{HX}$ in all the update rules can be substituted by the term $h(\mathbf{X})$, as argued in (Mandel, 2006). This is one of the principal advantages of the EnKF compared to the EKF and the Information filter, because it eliminates a potentially problematic computation of Jacobians and leads to more accurate *a posteriori* state estimates, as discussed in Section 2.6. Also, all the EnKF variants presented in this chapter do not change the asymptotic computational complexity of the EnKF update rule, which was presented in Table 2.4; all the costs still scale linearly with the number of state-space dimensions n.

The new update rules derived in this chapter, in particular the CPI-EnKF and the Augmented EnKF update rule, are not exclusive to the EnKF, and in principle their equivalents are applicable with other variants of the Kalman filter, such as the EKF or UKF, assuming that the cross-covariance between the state estimate error and the observation error is known. However, in the context of the EnKF, this cross-covariance can be estimated easily from samples, while with all other variants of the Kalman filter it needs to be maintained analytically, which is problematic, in particular in sensor networks as discussed in Chapter 4. In order to preserve the cross-covariance statistic between the state estimate and observation ensembles, the order of ensemble members must be retained, as discussed in Section 5.2.

In the context of distributed data fusion in sensor networks, the new EnKF update rules developed in this chapter can be potentially applied in very much the same way as the Covariance Intersection (CI) update rule is applied in the distributed data fusion algorithm described in Section 4.6. A practical example of such a data fusion system is presented in Section 7.1.3,

in which the CPI-EnKF and the Augmented EnKF filters are used to address the problem of cooperative localisation in a group of communicating vehicles.

5.3 Chapter Summary

This chapter presented the main contribution of this thesis - a generalisation of the EnKF to correlated estimation problems. Using this generalisation, the prediction step of the filter can provide consistent state estimates in the presence of any type of correlation, while avoiding the increase in the computational cost associated with the traditional state augmentation procedure. For the update step of the filter, this chapter derives three alternative update rules that can potentially operate in the presence of a correlation between the state estimate error and the observation error, each of whom is based on different presumptions. A numerical analysis of applicability of all the methods discussed in this chapter to the various correlated estimation problems is provided in Chapter 6. Chapter 7 then presents several practical examples of how such a framework can be applied to real-world correlated estimation problems.

Chapter 6

Numerical Evaluation

This chapter presents an evaluation of the performance of the EnKF modifications provided in Chapter 5 applied to various correlated estimation problems. The main goal of the evaluation is to assess to which types of problems the new filters are applicable, and to compare the quality of their estimates with the estimates provided by state-of-the-art methods. More specifically, four variants of the EnKF will be compared: the conventional EnKF, the Generalised EnKF, the CPI-EnKF and the Augmented EnKF. In the first part of the evaluation, the filters are used to estimate the state of several simulated one-dimensional physical systems affected by the types of correlation in the noises described in Chapters 3. In the second part, the filters are applied in the context of sensor networks in order to assess their applicability to the common past information problem, presented in Chapter 4.

Every evaluation scenario defines a specific model of a physical system, whose time evolution and observations are randomly simulated on a computer. In every step of such a simulation, each of the evaluated filters maintains an estimate of the state of the physical system, and the quality of the estimates is assessed with respect to the known true state. The main interest of the evaluation lies in the average performance of the filters, including the assessment of the consistency of the reported errors with the true errors. Therefore, the simulation of every evaluation scenario is repeated 10000 times with a different random seed in every run, and the quality metrics of filter estimates are aggregated over all the simulation runs. Such a high number of samples has an additional benefit that the confidence intervals around the results do not need to be shown in the graphs, because they are practically identical with the actual results and as such they do not provide any interesting information.

Formally, at every simulation time step each evaluated filter is characterised by two metrics:

the *true error* and the *reported error*. The true error, denoted as $\sigma_{\text{true}}(t) \in \mathbb{R}$, represents the root mean square (RMS) error of the filter's estimates at a time step t aggregated over all the simulation runs, formally:

$$\sigma_{\text{true}}(t) \;=\; \sqrt{\frac{1}{M}\sum_{i=1}^{M}\big[\mathbf{x}_i(t)-\hat{\mathbf{x}}_i(t)\big]^{\mathrm{T}}\big[\mathbf{x}_i(t)-\hat{\mathbf{x}}_i(t)\big]} \tag{6.1}$$

where $\mathbf{x}_i(t) \in \mathbb{R}^n$ denotes the true system state at the time step t in the ith simulation run, $\hat{\mathbf{x}}_i(t) \in \mathbb{R}^n$ denotes the mean of the corresponding state estimate provided by the particular filter, and $M \in \mathbb{N}$ represents the total number of simulations performed, herein $M = 10000$. The reported error, denoted as $\sigma_{\text{rep}}(t) \in \mathbb{R}$, represents the RMS standard deviation of the error reported by the filter, formally:

$$\sigma_{\text{rep}}(t) \;=\; \sqrt{\frac{1}{M}\sum_{i=1}^{M}\operatorname{tr}(\mathbf{P}_i(t))} \tag{6.2}$$

where $\mathbf{P}_i(t)$ denotes the estimated error covariance matrix associated with the filter's estimate at the time step t in the ith simulation run, and $\operatorname{tr}(\mathbf{P}_i(t))$ denotes the trace of this covariance matrix. From the definition of consistency provided in Equation (2.13), it follows that if a filter is consistent, then for every time step t:

$$\lim_{M\to\infty}\big[\,\sigma_{\text{rep}}(t)-\sigma_{\text{true}}(t)\,\big] \;\geq\; 0 \tag{6.3}$$

For overpessimistic filters, the inequality will be sharp ($>$), while for optimal filters, the inequality is actually an equality ($=$).

With the metrics described in the previous paragraph, the quality of the filters evaluated in this chapter can be assessed from the presented graphs and tables as follows. The lower the true error, the more accurate is the filter. If the reported error is significantly lower than the true error, the filter is not consistent. If it is significantly larger, the filter is overpessimistic. Good filters will have the reported error value very much the same as the true error.

6.1 No Correlation

As discussed in Section 2.6, the EnKF is a Monte Carlo method, in the sense that its results rely on repeated random sampling, and as such, the quality of the results depends on the number of random samplings performed. In other words, the higher the number of ensemble members used by the EnKF filter, the more accurate estimates the filter provides. On the other hand, the

memory and time requirements of the filter also grow with the number of ensemble members, and it is therefore necessary to compromise in the choice of the number of ensemble members with respect to the specific application. This section has two goals. First, to analyse from a practical perspective what is the relation between the number of ensemble members and the accuracy of the estimates provided by the EnKF variants. Second, to demonstrate that all the modifications of the EnKF presented in Chapter 5 are indeed generalisations of the EnKF, so that they provide equivalent estimates as the conventional EnKF in problems affected by no correlation.

For the purposes of this experiment, the following static one-dimensional system is used:

$$\mathbf{x}(t) \;=\; 0 \tag{6.4}$$

$$\mathbf{z}(t) \;=\; \mathbf{x}(t) + \mathbf{v}(t) \tag{6.5}$$

where $\mathbf{x}(t) \in \mathbb{R}$ represents the true system state at a time step t, and $\mathbf{z}(t) \in \mathbb{R}$ is an observation of the system that is affected by an independent white Gaussian noise $\mathbf{v}(t) \in \mathbb{R}$ with a covariance $\mathbf{R} = 1$. The state of the system was estimated using each of the EnKF variants described in Chapter 5, as well as the conventional EnKF described in Chapter 2. The experiments were repeated with the following ensemble sizes: $N \in \{100, 1000, 10000, 100000, 1000000\}$. Ground-truth estimates were computed using the conventional Kalman filter. The simulation was repeated 10000 times, as discussed in the introduction of this chapter. In every ith simulation run, all the filters were initialised with the same consistent estimate of the true system state, randomly generated as follows:

$$\hat{\mathbf{x}}_i(0) \;\sim\; \mathcal{N}(\,\mathbf{x}(0),\, \mathbf{P}_i(0)\,) \tag{6.6}$$

$$\mathbf{P}_i(0) \;=\; 100 \tag{6.7}$$

The filter statistics were collected after their update steps.

The graphs in Figures 6.1 and 6.2 show the evolution of the filters' estimates over time in cases of $N = 100$ and $N = 1000$, respectively. In the case of $N = 100$, the conventional EnKF is practically equivalent to the ground-truth Kalman filter, while the modifications of the EnKF have noticeably larger errors, in particular the Generalised EnKF. However, these errors diminish as the number of ensemble members increases to $N = 1000$. The Augmented EnKF provides practically identical results as the CPI-EnKF, which stems from the fact that for one-dimensional problems the filters are algebraically equivalent. Table 6.1 presents a more detailed view of the accuracy of the estimates provided by all the filters at a snapshot time step

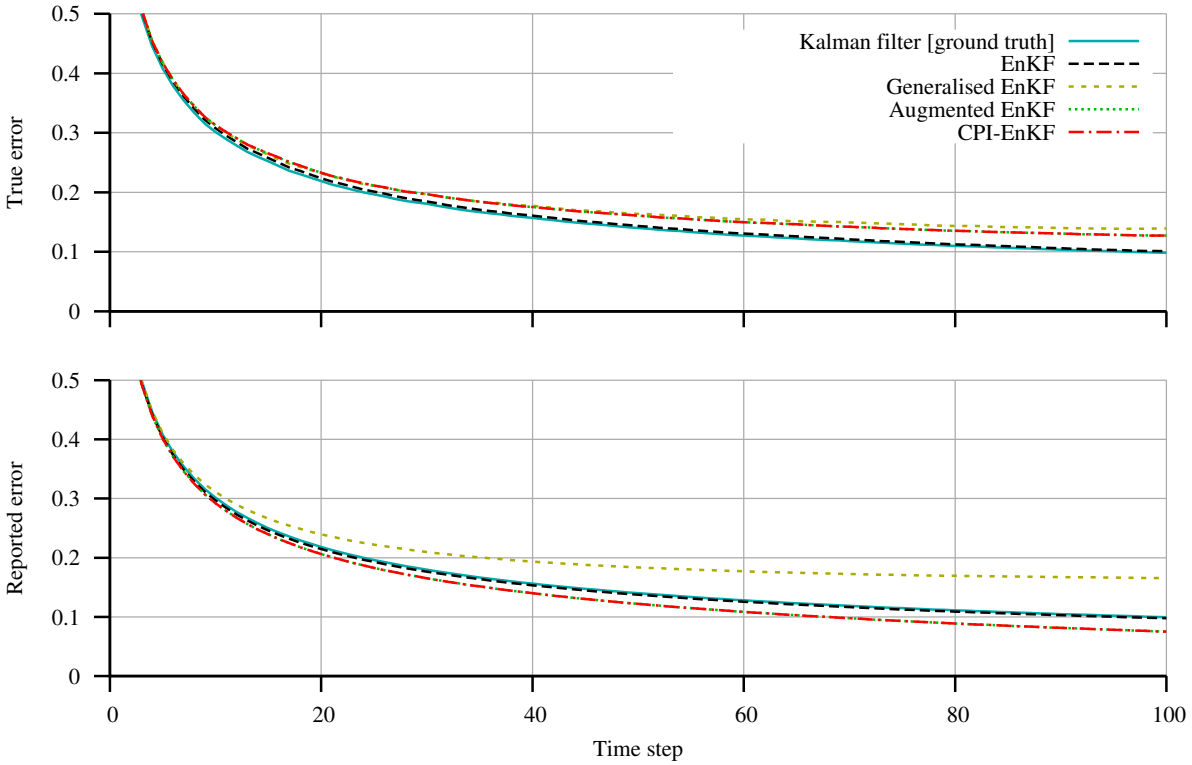


Fig. 6.1: Accuracy of EnKF variants with $N = 100$, no correlation.

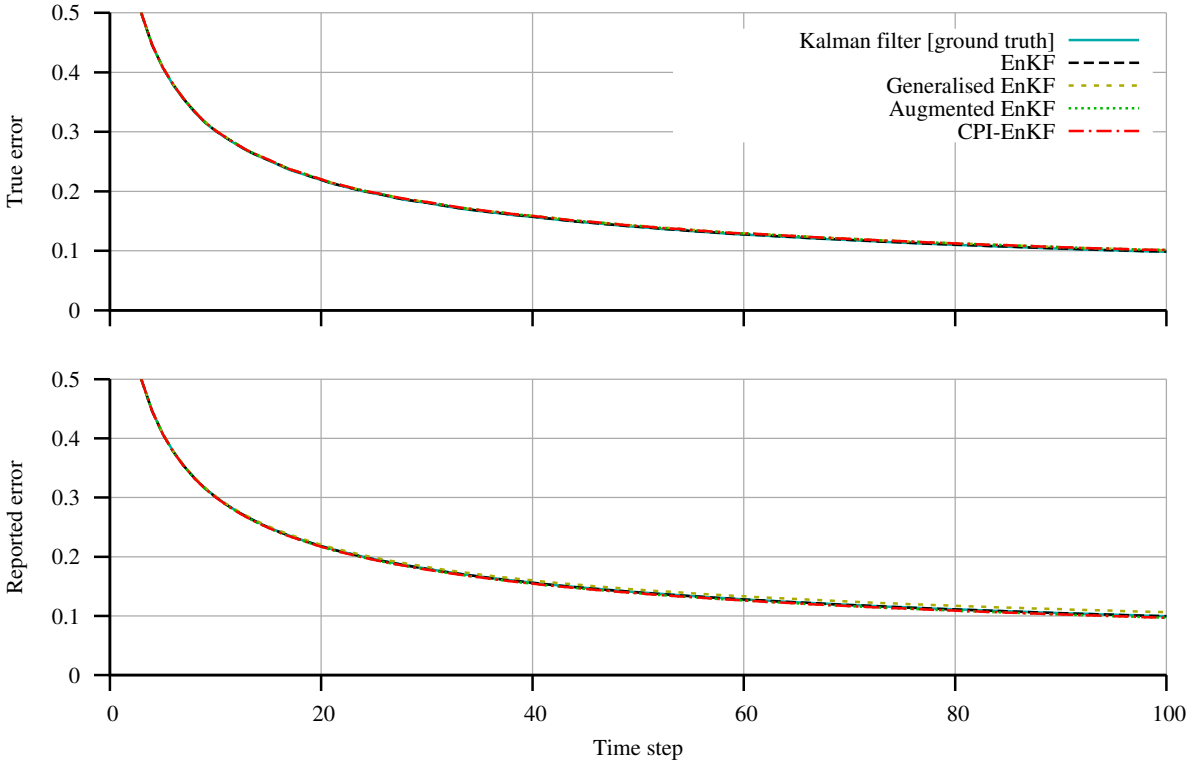


Fig. 6.2: Accuracy of EnKF variants with $N = 1000$, no correlation.

	$N = 10^2$	$N = 10^3$	$N = 10^4$	$N = 10^5$	$N = 10^6$	
Kalman filter	0.21882					True error
Basic EnKF	0.22377	0.21934	0.21893	0.21883	0.21882	
Generalised EnKF	0.23176	0.22011	0.21892	0.21889	0.21882	
CPI-EnKF	0.23304	0.22027	0.21911	0.21879	0.21882	
Augmented EnKF	0.23293	0.22022	0.21910	0.21880	0.21882	
Kalman filter	0.21816					Reported error
Basic EnKF	0.21458	0.21778	0.21814	0.21815	0.21816	
Generalised EnKF	0.23953	0.22029	0.21839	0.21818	0.21816	
CPI-EnKF	0.20618	0.21694	0.21806	0.21814	0.21816	
Augmented EnKF	0.20620	0.21694	0.21806	0.21814	0.21816	

Table 6.1: Accuracy of EnKF variants with different ensemble sizes, no correlation. The values in the table represent a snapshot of all the simulations at time step $t = 20$. Note that the red digits indicate where the true error becomes greater than ground truth (Kalman filter), or where the reported error becomes lower than the corresponding true error.

$t = 20$. Clearly, the accuracy of all the filters increases with the increasing number of ensemble members, as the estimates of all the EnKF variants converge towards the ground-truth Kalman filter estimate. The results indicate that all the filters provide consistent estimates.

In summary, the experiments presented in this section demonstrate that all the variants of the EnKF described in Chapter 5 approximate the ground-truth estimates provided by the Kalman filter. The Generalised EnKF, the Augmented EnKF and the CPI-EnKF, however, require higher number of ensemble members to match the accuracy of the conventional EnKF.

6.2 Non-Linear Non-Gaussian Dynamical System

Although the EnKF depends on a theoretical assumption that the underlying dynamical system is linear, the filter is routinely being applied to problems where such an assumption is not satisfied, as discussed in Section 2.6, and it can even provide more accurate state estimates in this situation compared to other variants of the Kalman filter, such as the EKF and UKF. The experiment presented in this section evaluates whether the new generalisations of the EnKF developed in this thesis retain this property, using a well-known non-linear non-Gaussian dynamical system

originally proposed in (Kitagawa, 1987):

$$\mathbf{x}(t) \;=\; 0.5\,\mathbf{x}(t-1) \;+\; \frac{25\,\mathbf{x}(t-1)}{1+\mathbf{x}^2(t-1)} \;+\; 8\cos(1.2(t-1)) \;+\; \mathbf{w}(t) \tag{6.8}$$

$$\mathbf{z}(t) \;=\; 0.05\,\mathbf{x}^2(t) \;+\; \mathbf{v}(t) \tag{6.9}$$

where $\mathbf{x}(t) \in \mathbb{R}$ represents the true system state at a time step t, $\mathbf{w}(t) \in \mathbb{R}$ denotes a white process noise with covariance $\mathbf{Q} = 10$, and $\mathbf{z}(t) \in \mathbb{R}$ represents an observation of the system, affected by a white observation noise with covariance $\mathbf{R} = 1$.

The state of the non-linear dynamical system was estimated using the EKF, conventional EnKF, the Generalised EnKF, the CPI-EnKF and the Augmented EnKF. All the EnKF variants used the same number of ensemble members $N = 1000$. In every ith simulation run, all the filters were initialised with the same consistent estimate of the true system state, generated randomly as follows:

$$\hat{\mathbf{x}}_i(0) \;\sim\; \mathcal{N}(\,\mathbf{x}_i(0),\, \mathbf{P}_i(0)\,) \tag{6.10}$$

$$\mathbf{P}_i(0) \;=\; 100 \tag{6.11}$$

where $\mathbf{x}_i(0) \in \mathbb{R}$ denotes the initial true system state in the ith simulation, which is also generated randomly as:

$$\mathbf{x}_i(0) \;\sim\; \mathcal{N}(\,0,\, \mathbf{Q}\,) \tag{6.12}$$

In this experiment, the filter statistics were collected after their update steps.

The results of the experiment are presented in Figure 6.3. Clearly, all the variants of the EnKF provide practically identical estimates with a much higher accuracy than the EKF. Moreover, unlike the EKF, their estimates appear to be consistent with the reported errors.

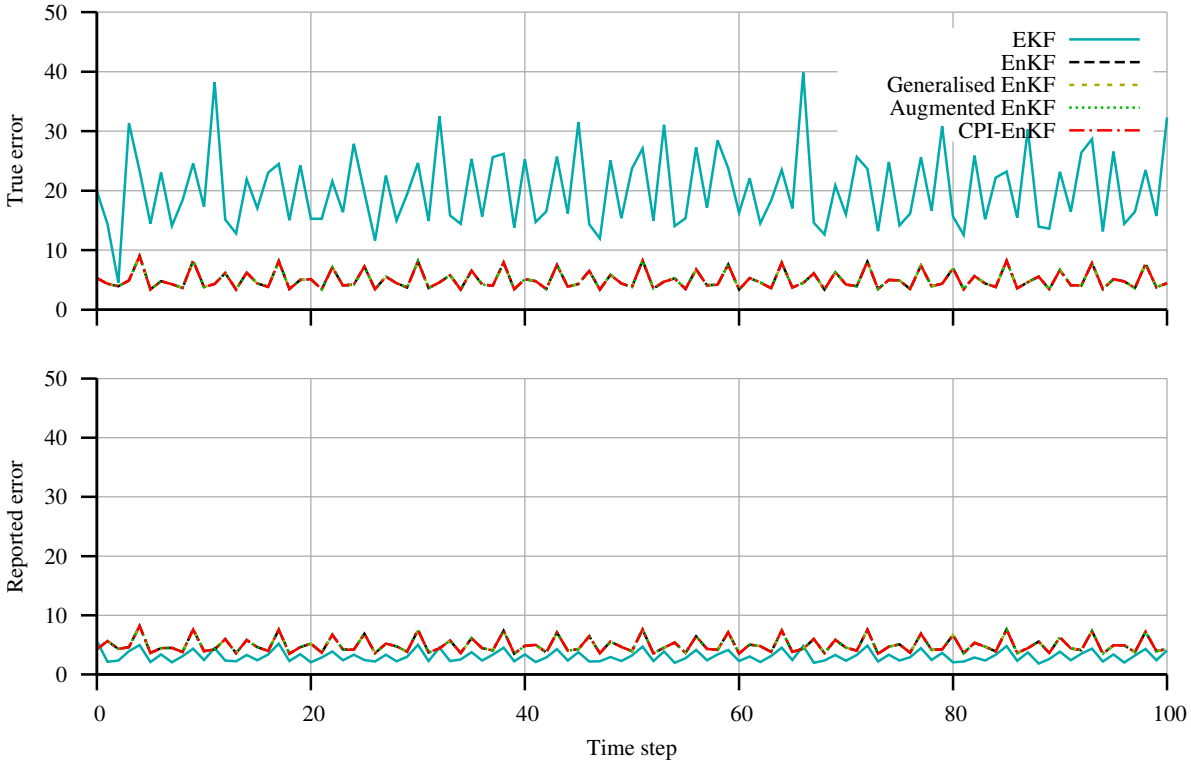


Fig. 6.3: Accuracy of EnKF variants, non-linear non-Gaussian dynamical system.

6.3 Correlated Noises

In this section, the three variants of the EnKF described in Chapter 5 are applied to estimation problems where the process and observation noises are affected by the types of correlation described in Chapter 3. More specifically, the following types of correlation in the noises are analysed: a sequentially correlated process noise, a correlated process and observation noise and a sequentially correlated observation noise.

6.3.1 Sequentially Correlated Process Noise

The goal of this experiment is to evaluate the performance of the EnKF variants in an estimation problem affected by a sequentially correlated process noise, a problem which has been discussed in Section 3.2. For the purpose of this experiment, the following dynamical system was used:

$$\mathbf{x}(t) = 0.9\,\mathbf{x}(t-1) + \mathbf{w}(t) \tag{6.13}$$

$$\mathbf{z}(t) = \mathbf{x}(t) + \mathbf{v}(t) \tag{6.14}$$

where $\mathbf{x}(t) \in \mathbb{R}$ represents the true system state at a time step t, $\mathbf{w}(t) \in \mathbb{R}$ denotes a coloured process noise with covariance $\mathbf{Q} = 1$, and $\mathbf{z}(t) \in \mathbb{R}$ represents an observation of the system, which is affected by a white observation noise $\mathbf{v}(t) \in \mathbb{R}$ with covariance $\mathbf{R} = 1$. The process noise sequence $\{\mathbf{w}(t)\}$ is a coloured Gaussian noise process, as described in Definition 3.3, modelled by the following equation:

$$\mathbf{w}(t+1) \;=\; \boldsymbol{\Psi}\,\mathbf{w}(t) \;+\; \varepsilon(t) \tag{6.15}$$

where $\boldsymbol{\Psi} \in \mathbb{R}$ is an auto-correlation coefficient whose value varies between the experiments, and $\varepsilon(t) \in \mathbb{R}$ represent an independent white Gaussian noise with covariance $\boldsymbol{\Sigma} = 0.9\,(1 - \boldsymbol{\Psi}^2)$.

The state of the system is estimated using the EnKF, with the process noise compensation term in the prediction step generated according to the process noise model from Equation (6.15), using a method that has been described in Section 5.1. The EnKF has a fixed number of ensemble members $N = 1000$. The ground-truth estimates are provided by a Kalman filter with augmented state, as described in Section 3.2. For a reference, the system state is also estimated (incorrectly) using a conventional Kalman filter, as well as the conventional EnKF with an augmented state. In every ith simulation run, all the filters are initalised with the same consistent estimate of the true system state, generated randomly as follows:

$$\hat{\mathbf{x}}_i(0) \;\sim\; \mathcal{N}(\,\mathbf{x}_i(0),\, \mathbf{P}_i(0)\,) \tag{6.16}$$

$$\mathbf{P}_i(0) \;=\; 100 \tag{6.17}$$

where $\mathbf{x}_i(0) \in \mathbb{R}$ denotes the initial true system state in the ith simulation, which is also generated randomly as:

$$\mathbf{x}_i(0) \;\sim\; \mathcal{N}(\,0,\, \mathbf{Q}\,) \tag{6.18}$$

In this experiment, the filter statistics were collected after their prediction steps.

The graphs in Figures 6.4 and 6.5 show the performance of the filters in a scenario with the auto-correlation coefficient set as $\boldsymbol{\Psi} = 0.5$ and $\boldsymbol{\Psi} = 0.99$, respectively. Clearly, the EnKF with correlated process noise compensation is sub-optimal, although still consistent. As expected, the EnKF with augmented state is practically equivalent to the ground-truth Kalman filter with augmented state, and the conventional Kalman filter provides inconsistent estimates. The conclusion of these experiment is that although the EnKF with correlated process noise compensation provides consistent estimates, in practical applications it is better to use the traditional state augmentation method described in Section 3.2. With the EnKF, the performance penalty associated with such state augmentation is typically manageable.

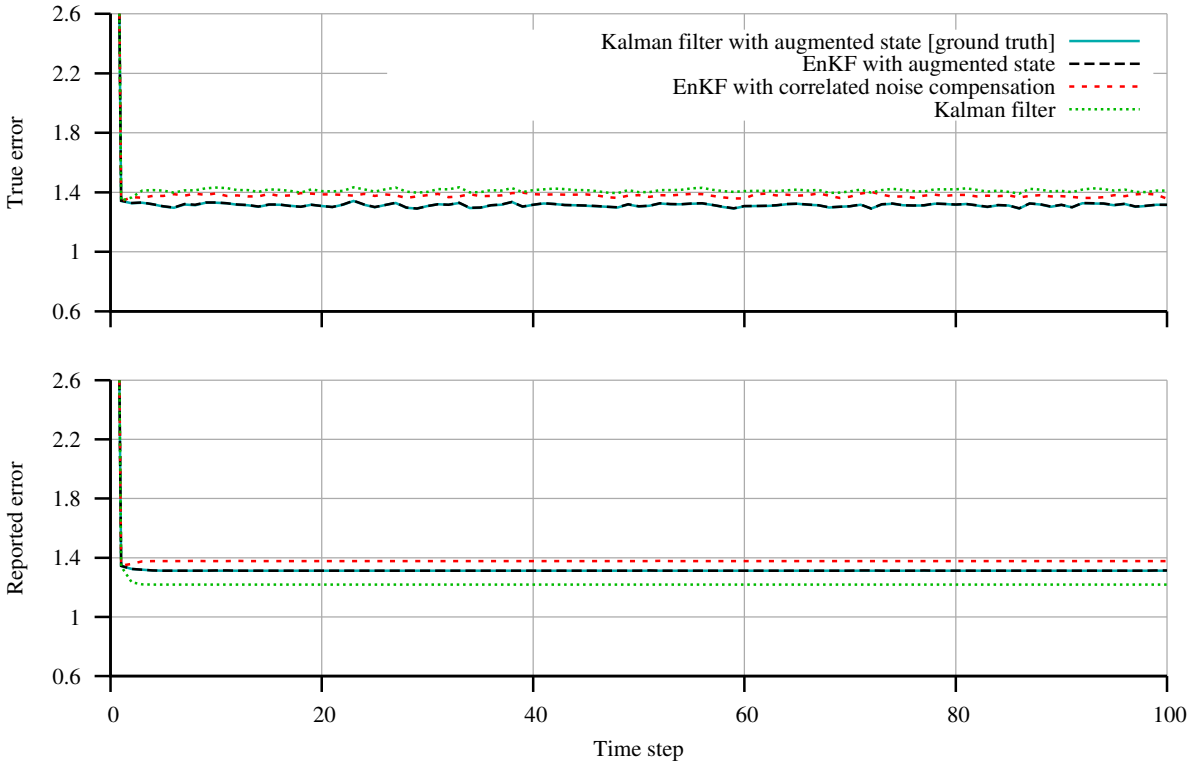


Fig. 6.4: Accuracy of EnKF variants, sequentially correlated process noise with $\mathbf{\Psi} = 0.5$.

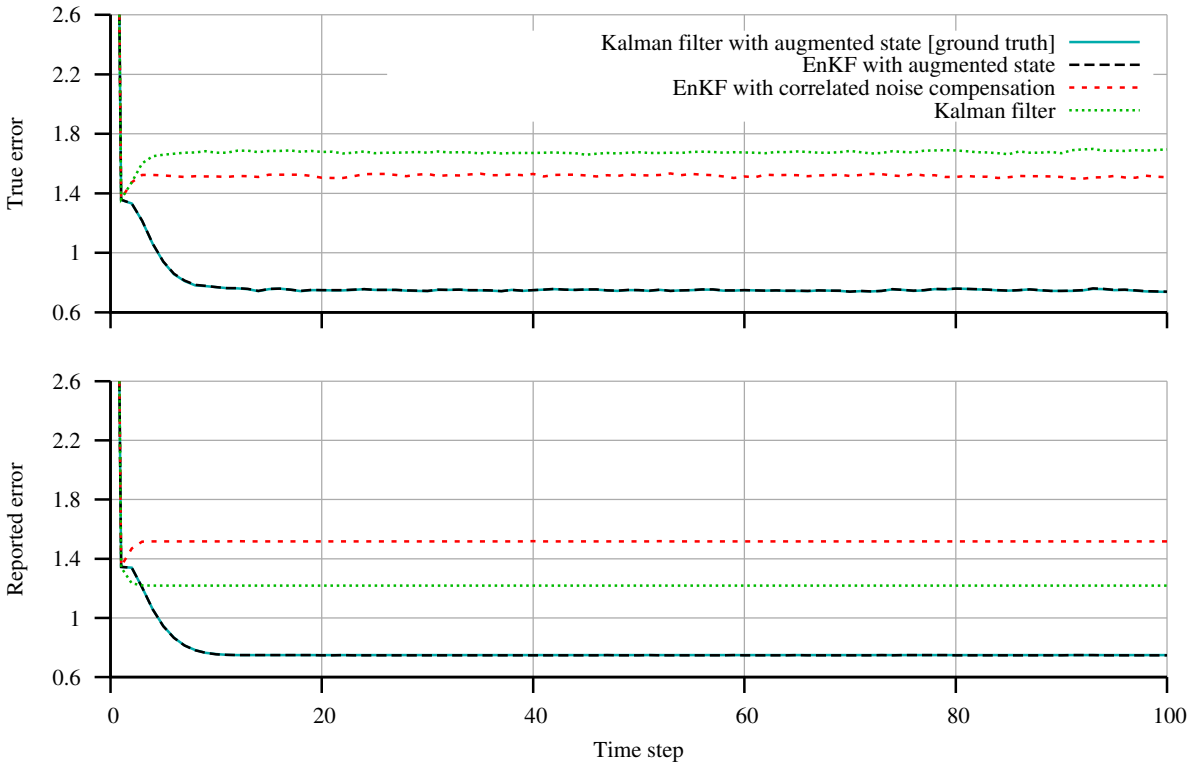


Fig. 6.5: Accuracy of EnKF variants, sequentially correlated process noise with $\mathbf{\Psi} = 0.99$.

6.3.2 Correlated Process and Observation Noise

The experiments presented in this section aim to evaluate whether any of the variants of the EnKF presented in Chapter 5 is applicable to estimation problems in which the process and observation noises are correlated, as discussed in Section 3.1. For the purpose of this experiment, the same one-dimensional dynamical system as in the previous section was used:

$$\mathbf{x}(t) \;=\; 0.9\,\mathbf{x}(t-1) + \mathbf{w}(t) \tag{6.19}$$

$$\mathbf{z}(t) \;=\; \mathbf{x}(t) + \mathbf{v}(t) \tag{6.20}$$

and the process and observation noise covariances are also defined as $\mathbf{Q} = 1$ and $\mathbf{R} = 1$, respectively. Both the process noise sequence $\{\mathbf{w}(t)\}$ and the observation noise sequence $\{\mathbf{v}(t)\}$ represent white Gaussian noise processes, which are mutually correlated as follows:

$$\mathrm{cov}(\,\mathbf{w}(i), \mathbf{v}(j)\,) = \begin{cases} \mathbf{C} & i = j \\ 0 & i \neq j \end{cases} \tag{6.21}$$

where $\mathbf{C} \in \mathbb{R}$ is a variable depending on the scenario.

The state of the system was estimated using the conventional EnKF, the Generalised EnKF, the CPI-EnKF and the Augmented EnKF. All these filters used the same number of ensemble members $N = 1000$, and they all had the noise compensation terms generated according to the model in Equation (6.21), using a method described in Section 5.2, i.e. the state estimate and observation ensembles are mutually correlated prior to an update as $\mathrm{cov}(\mathbf{X}, \mathbf{Z}) \simeq \mathbf{C}$. The ground-truth estimates are computed by the Generalised Kalman filter from Definition 3.1, and for a reference, the system is also estimated (incorrectly) using the conventional Kalman filter. Similarly to the previous section, all the filters are initialised with the same consistent estimate, which is generated randomly in every ith simulation run as follows:

$$\hat{\mathbf{x}}_i(0) \;\sim\; \mathcal{N}(\,\mathbf{x}_i(0),\, \mathbf{P}_i(0)\,) \tag{6.22}$$

$$\mathbf{P}_i(0) \;=\; 100 \tag{6.23}$$

where $\mathbf{x}_i(0) \in \mathbb{R}$ denotes the initial true system state in the ith simulation, which is also generated randomly as:

$$\mathbf{x}_i(0) \;\sim\; \mathcal{N}(\,0,\, \mathbf{Q}\,) \tag{6.24}$$

The filter statistics were collected after the update steps.

The results of the experiment are presented in Figures 6.6 and 6.7, for correlation coefficient defined as $\mathbf{C} = 0.5$ and $\mathbf{C} = 0.99$, respectively. Note that for one-dimensional dynamical

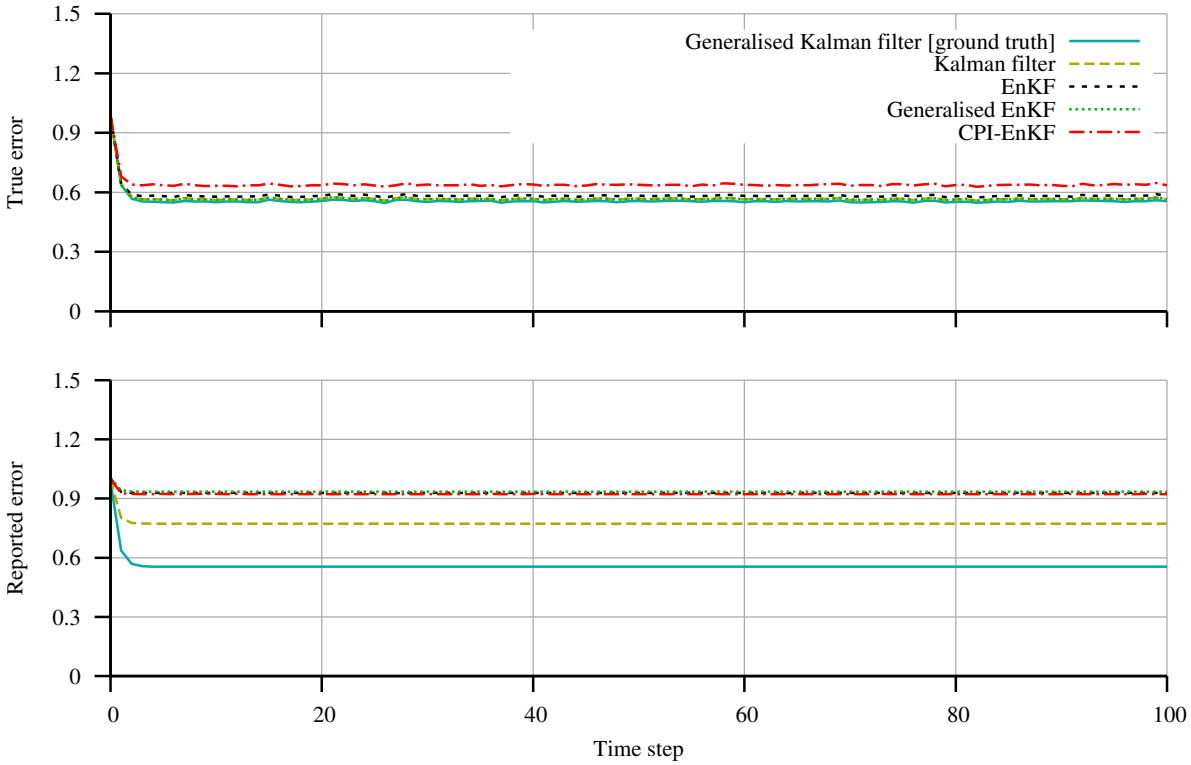


Fig. 6.6: Accuracy of EnKF variants, correlated process and observation noise with $\mathbf{C} = 0.5$.

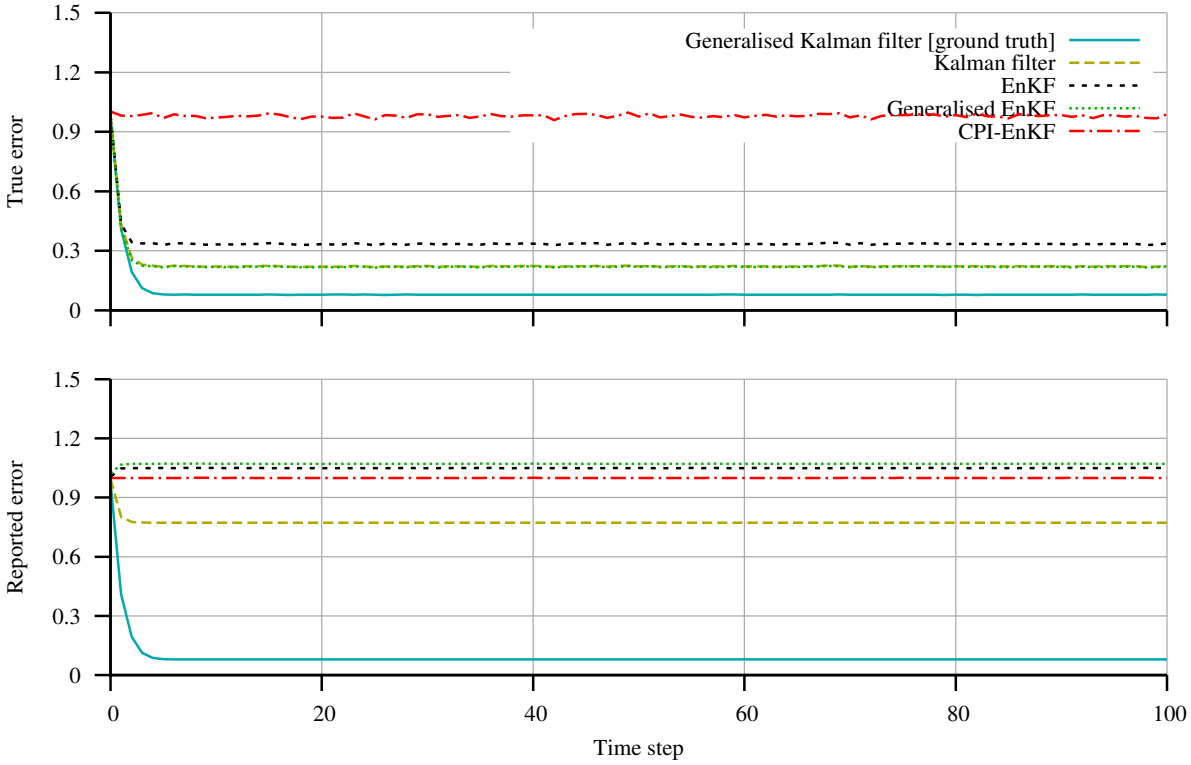


Fig. 6.7: Accuracy of EnKF variants, correlated process and observation noise with $\mathbf{C} = 0.99$.

systems, the CPI-EnKF and the Augmented EnKF are algebraically equivalent, so for clarity, the graphs in this section only include the CPI-EnKF. In summary, although all the EnKF variants appear to provide consistent estimates, their error is very large. As such, it can be said that the EnKF is **not** suitable for estimation problems where the process and observation noises are correlated and the correlation is known. Also, the fact that the Generalised EnKF provides very different results compared to the Generalised Kalman filter is effectively a counter-example to the Generalised EnKF. In other words, this experiment demonstrates that the Generalised EnKF is not a valid filter, and as such, it should not be used in practical applications.

6.3.3 Sequentially Correlated Observation Noise

This experiment aims to evaluate the performance of the EnKF variants in an estimation problem affected by a sequentially correlated observation noise, a problem which has been discussed in Section 3.3. For this purpose, the following static one-dimensional system is used:

$$\mathbf{x}(t) \;=\; 0 \tag{6.25}$$

$$\mathbf{z}(t) \;=\; \mathbf{x}(t) + \mathbf{v}(t) \tag{6.26}$$

The observation noise $\mathbf{v}(t) \in \mathbb{R}$ has covariance $\mathbf{R} = 1$, and the sequence $\{\mathbf{v}(t)\}$ forms a coloured Gaussian noise process, modelled by the following equation:

$$\mathbf{v}(t+1) \;=\; \mathbf{\Psi}\,\mathbf{v}(t) \;+\; \varepsilon(t) \tag{6.27}$$

where $\mathbf{\Psi} \in \mathbb{R}$ is a variable auto-correlation coefficient depending on the evaluation scenario, and $\varepsilon(t) \in \mathbb{R}$ is an independent white Gaussian noise with covariance $\mathbf{\Sigma} = 1 - \mathbf{\Psi}^2$.

The state of the system is estimated using the conventional EnKF, the Generalised EnKF, the CPI-EnKF and the Augmented EnKF. All of these filters have the same number of ensemble members $N = 1000$, and they all have the observation noise compensation generated according to the observation noise model from Equation (6.27), using a method discussed in Section 5.2. The ground-truth estimates are computed using a Kalman filter with augmented state as described in Section 3.3. As a reference, the system state is also estimated (incorrectly) using a conventional Kalman filter. In every ith simulation, all the filters were initalised with the same consistent estimate of the true system state, randomly generated as follows:

$$\hat{\mathbf{x}}_i(0) \;\sim\; \mathcal{N}(\,\mathbf{x}(0),\, \mathbf{P}_i(0)\,) \tag{6.28}$$

$$\mathbf{P}_i(0) \;=\; 100 \tag{6.29}$$

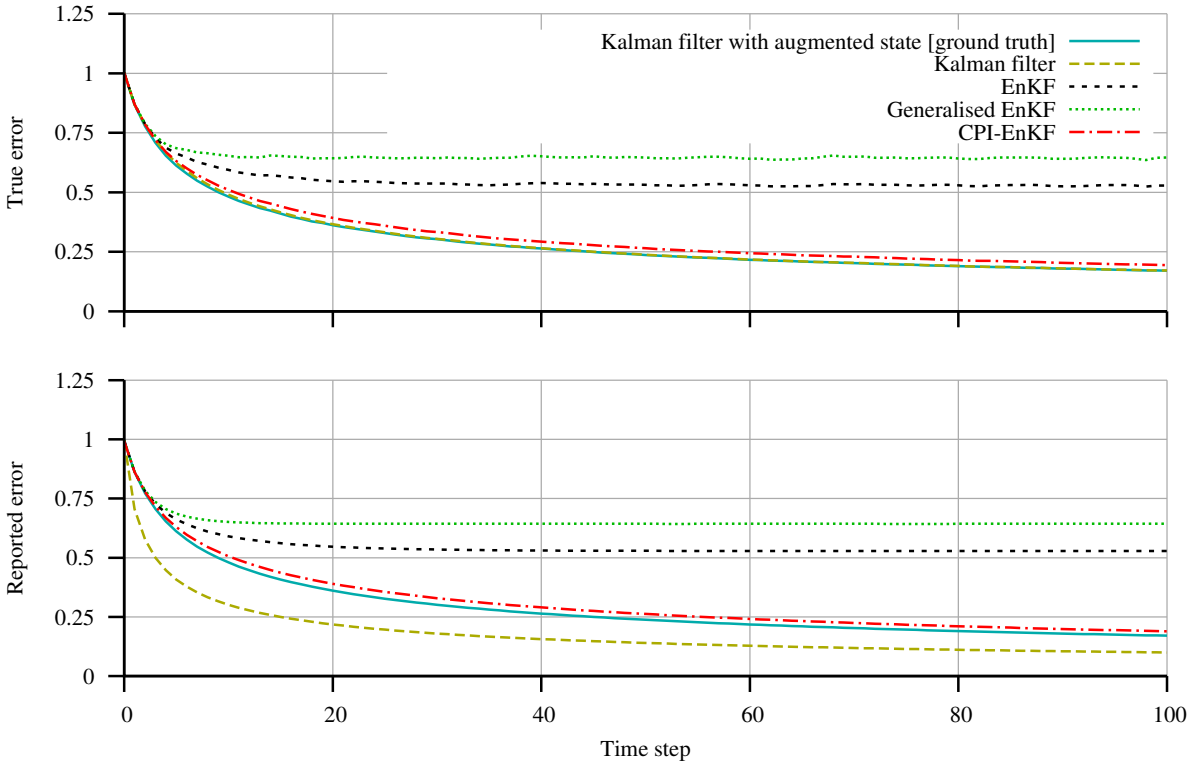


Fig. 6.8: Accuracy of EnKF variants, correlated observation noise with $\mathbf{\Psi} = 0.5$.

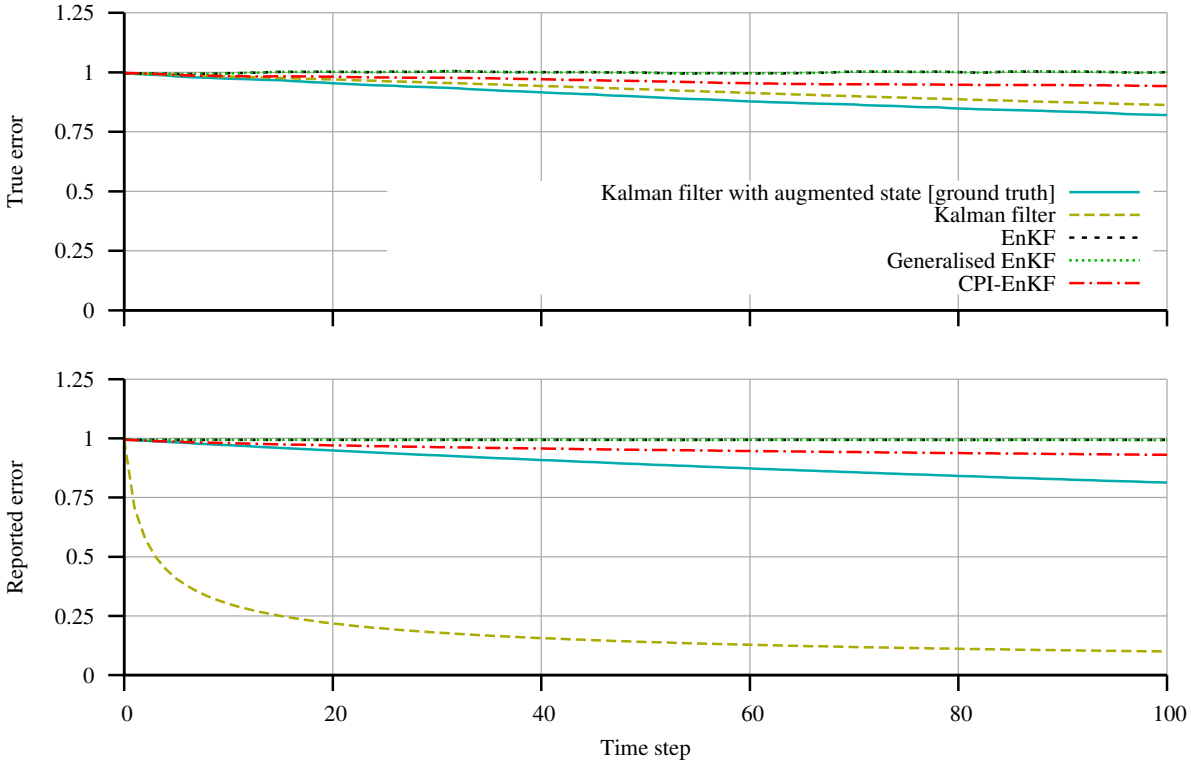


Fig. 6.9: Accuracy of EnKF variants, correlated observation noise with $\mathbf{\Psi} = 0.99$.

In this experiment, the filter statistics were collected after their update steps.

Figures 6.8 and 6.9 show the performance of the filters in a scenario with the auto-correlation coefficient set as $\mathbf{\Psi} = 0.5$ and $\mathbf{\Psi} = 0.99$, respectively. As similarly noted in the previous section, in one-dimensional systems the CPI-EnKF and the Augmented EnKF are algebraically equivalent, so for clarity, the graphs only include the CPI-EnKF. The results indicate that both the conventional EnKF and the Generalised EnKF provided estimates with a rather high error, although still consistent. On the other hand, the CPI-EnKF (and hence the Augmented EnKF) provides state estimates with a very reasonable error that is only slightly worse than the ground truth, and importantly, the results indicate that the corresponding error covariance estimates are consistent. As expected, the conventional Kalman filter provides inconsistent estimates.

In summary, the experiments performed in this section indicate that the CPI-EnKF and Augmented EnKF can be applied to estimation problems with a sequentially correlated observation noise with good accuracy. Although the conventional state augmentation and measurement differencing methods, which are described in Section 3.3, provide potentially more accurate state estimates, in particular if the sequential correlation is of a high magnitude, the CPI-EnKF (and Augmented EnKF) avoids certain problems of these methods, such as the increased computational complexity or a lag-one delay in estimates. This feature represents a significant advantage of the CPI-EnKF in certain applications, for example, the application presented in Section 7.3.

6.4 The Common Past

This section provides an evaluation of applicability of the EnKF variants, described in Chapter 5, to estimation problems where the correlations between the state estimate errors and the observation errors are caused by the presence of common past information shared between the two - a problem that has been discussed in detail in Chapter 4. As will be demonstrated, this is quite a different problem than the case where such a correlation was caused by a correlation in the noises, and therefore it requires a different treatment.

6.4.1 Shared Error Term

Consider the following problem. There is an estimate $\hat{\mathbf{x}} \in \mathbb{R}^n$ of a true state $\mathbf{x} \in \mathbb{R}^n$ of some physical system, and a direct observation $\mathbf{z} \in \mathbb{R}^n$ of the same system. Both the errors associated with the state estimate and the observation have a zero-mean Gaussian probability distribution, and both are composed of two zero-mean Gaussian components: a shared error component with

a covariance $\mathbf{E} \in \mathbb{R}^{n \times n}$ and two independent error components with covariances $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times n}$, formally:

$$\mathrm{cov}(\mathbf{x} - \hat{\mathbf{x}}) = \mathbf{E} + \mathbf{A} \tag{6.30}$$

$$\mathrm{cov}(\mathbf{x} - \mathbf{z}) = \mathbf{E} + \mathbf{B} \tag{6.31}$$

As such, the state estimate error and the observation error are mutually correlated as:

$$\mathrm{cov}(\mathbf{x} - \hat{\mathbf{x}}, \mathbf{x} - \mathbf{z}) = \mathbf{E} \tag{6.32}$$

The shared error term with covariance $\mathbf{E}$ represents common past information shared between the state estimate and the observation. In an ideal fusion system, such a shared error would be ignored altogether during assimilation of the observation, and the resulting state estimate would only be based on covariances $\mathbf{A}$ and $\mathbf{B}$. Note that the error model described above precisely matches the formal assumptions of the CPI-EnKF described in Section 5.2.2. The goal of the experiment performed in this section is to evaluate the performance of the CPI-EnKF on a canonical instance of such an estimation problem, as well as the performance of the other variants of the EnKF and all the applicable state-of-the-art methods, including the Generalised Kalman filter (Generalised KF), the Covariance Intersection (CI) and the Split Covariance Intersection (Split CI) algorithms, which have been described in Section 4.6.

The evaluation scenario comprises a two-dimensional system with the following parameters:

$$\mathbf{x} = \begin{bmatrix} 10 \\ -5 \end{bmatrix} \quad \mathbf{E} = \begin{bmatrix} 3 & -3 \\ -3 & 5 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 3 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.33}$$

Note that the values were chosen semi-randomly so that different types of covariance matrices are included, such as a covariance matrix with negative cross-correlation ($\mathbf{E}$), a covariance matrix with positive cross-correlation ($\mathbf{A}$) and a diagonal covariance matrix ($\mathbf{B}$). In every simulation run, both the state estimate $\hat{\mathbf{x}} \in \mathbb{R}^2$ and the observation $\mathbf{z} \in \mathbb{R}^2$ are generated randomly according to this model, and then they are fused using each of the filters evaluated. The EnKF variants are evaluated in versions with the following ensemble sizes: $N \in \{100, 1000, 10000, 100000, 1000000\}$, and ground truth is provided by a conventional Kalman filter using the (hidden) covariances $\mathbf{A}$ and $\mathbf{B}$. The Generalised Kalman filter employs the shared error covariance $\mathbf{E}$ in the place of the covariance between the process and observation noise $\mathbf{C}$ (see Definition 3.1). Similarly, the Split CI employs $\mathbf{E}$ in the place of the known-correlated components of the state estimate error $\mathbf{P}_C^-(t)$ and the observation error $\mathbf{R}_C$ (see Definition 4.3).

The results of the experiment are presented in Table 6.2. One of the most important observations is that the estimates provided by the Generalised Kalman filter have not only a relatively high error, but also that they are inconsistent. Clearly, this implies that the Generalised Kalman filter is not applicable to such a shared error estimation problem, and **indicates that a mere knowledge of the covariance between the state estimate and observation errors is not sufficient information to perform consistent data fusion, without a valid interpretation of the origin of correlation**. Apparently, both the CI and the Split CI are conservative enough to be insensitive to such an interpretation, as they both provide consistent but overpessimistic estimates, although with a relatively high error. As expected, the conventional Kalman filter is inconsistent, and the conventional EnKF and the Generalised EnKF have a relatively high error. Finally, both the CPI-EnKF and the Augmented EnKF compute estimates equivalent to the ground truth, with an accuracy only limited by the number of ensemble members.

	$N = 10^2$	$N = 10^3$	$N = 10^4$	$N = 10^5$	$N = 10^6$	
Ground truth	1.22894					True error
Generalised KF	1.36607					
Kalman filter	1.33118					
CI	1.40787					
Split CI	1.36618					
EnKF	1.37308	1.33580	1.33129	1.33122	1.33118	
Generalised EnKF	1.40273	1.37023	1.36630	1.36613	1.36607	
CPI-EnKF	1.31974	1.23802	1.22917	1.22897	1.22895	
Augmented EnKF	1.33416	1.23932	1.22907	1.22892	1.22893	
Ground truth	3.08323					Reported error
Generalised KF	1.41196					
Kalman filter	2.43003					
CI	3.42741					
Split CI	3.16024					
EnKF	3.11298	3.12485	3.12623	3.12638	3.12640	
Generalised EnKF	3.13954	3.14106	3.14149	3.14155	3.14155	
CPI-EnKF	3.06320	3.08106	3.08298	3.08319	3.08321	
Augmented EnKF	3.05306	3.08008	3.08289	3.08318	3.08321	

Table 6.2: Applicability of various filters to the shared error term problem. Note that the reported errors are much larger than the true errors, which is due to the fact that they account for covariance of the shared error. Therefore, they should only be considered with relation to the ground-truth reported error. The red digits indicate where the true error becomes greater than ground truth, or where the reported error becomes lower than ground truth.

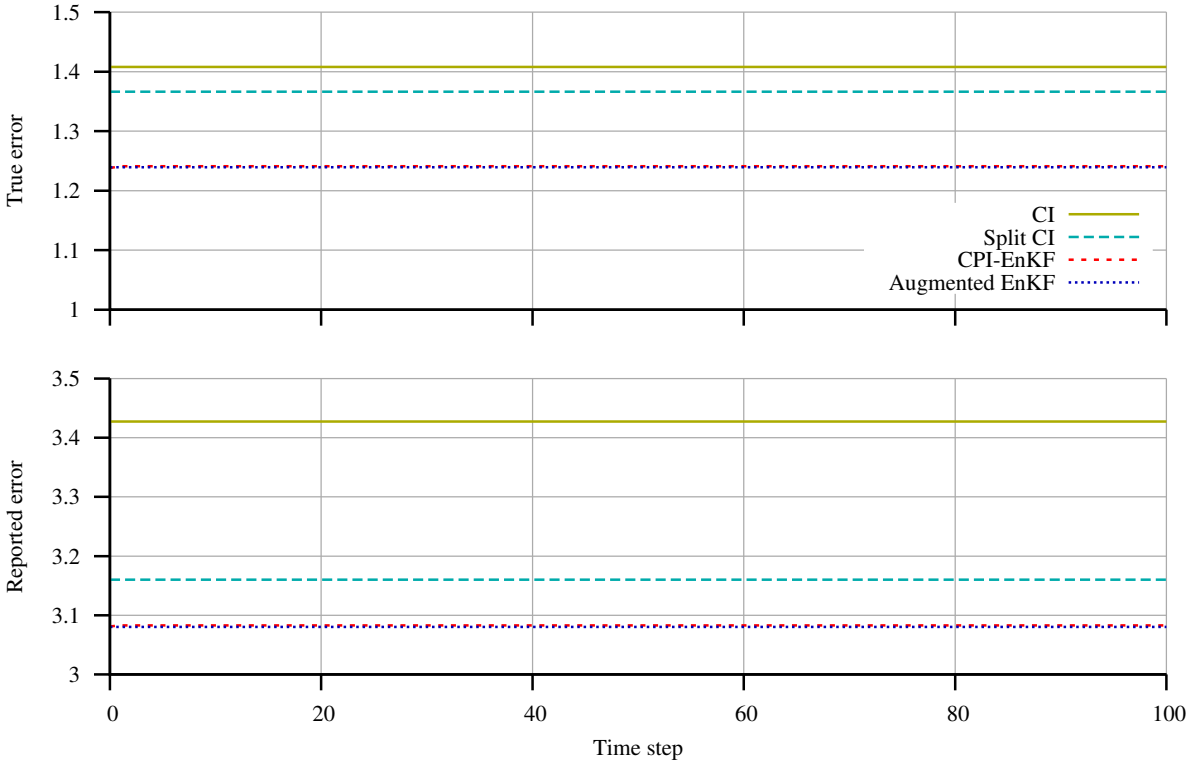


Fig. 6.10: Idempotence of EnKF variants with respect to the update operation.

6.4.2 Idempotence of Update

The goal of this section is to evaluate how the EnKF variants handle a situation where the same observation is repeatedly assimilated into the state estimate. An ideal filter should only alter the state estimate during the first such assimilation, but not during subsequent ones as they provide no new information. In order to evaluate such a property of the filter, which can also be called *idempotence* with respect to the update operation, the scenario from Section 6.4.1 is reused, and the same observation is repeatedly supplied to the filter. Figure 6.10 shows a time evolution of the average estimates provided by several of the filters in such a simulation. For clarity, the figure only includes the most relevant filters, which also provided consistent estimates already in Section 6.4.1: the CI, Split CI, CPI-EnKF and Augmented EnKF, the latter two with $N = 1000$. In summary, all these filters are idempotent with respect to the update operation.

6.4.3 Tree-Connected Sensor Network

In the final experiment presented in this chapter, the EnKF variants are employed to perform state estimation in a distributed sensor network. For the purposes of this experiment, the

following two-dimensional static system is used:

$$\mathbf{x}(t) = \begin{bmatrix} 10 \\ -5 \end{bmatrix} \qquad \mathbf{z}(t) = \begin{bmatrix} 1 & 0 \\ 2 & -1 \end{bmatrix} \mathbf{x}(t) + \mathbf{v}(t) \qquad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.34}$$

where $\mathbf{R}$ denotes the covariance of the observation noise $\mathbf{v}(t) \in \mathbb{R}^2$. Note that the values were chosen semi-randomly, so that the observation model is an invertible and asymmetric matrix.

Each node in the sensor network maintains its own local estimate of the system state and neighbouring nodes can exchange messages, as described in Chapter 4. The experiment involves three different types of network topology with 10 nodes, all of whom are depicted in Figure 6.11, including the numeric indices of the nodes. Note that all the topologies are tree-connected, so that objective ground-truth estimates can be computed using the Channel filters approach, described in Section 4.4. Note that a computation of the ground-truth estimates using the centralised Kalman filter or Decentralised Kalman filter approaches, described in Sections 4.1 and 4.2, respectively, would be unfair because they both assume instantaneous communication.

For clarity of presentation of the results, only the most relevant filters are included in this experiment: the Split CI, the Generalised EnKF, the CPI-EnKF and the Augmented EnKF. All these filters are applied in a similar way as the CI algorithm is applied in the generic data fusion algorithm described in Section 4.6. For each node k and simulation run i, the local estimate $\hat{\mathbf{x}}_{i|k}(0)$ and the associated error covariance $\mathbf{P}_{i|k}(0)$ of all the filters are initialised to the same value, which is generated randomly as follows:

$$\hat{\mathbf{x}}_{i|k}(0) \sim \mathcal{N}(\mathbf{x}(0), \mathbf{P}_{i|k}(0)) \tag{6.35}$$

$$\mathbf{P}_{i|k}(0) = 0.1(k+1) \tag{6.36}$$

The results of the evaluation for the path, star and balanced tree sensor network topology scenarios are presented in Figures 6.12, 6.13 and 6.13, respectively. In summary, the graphs indicate that all algorithms provide consistent state estimates. The Generalised EnKF provides estimates with a relatively high error in all three scenarios, which yet again indicates that it is not a consistent filter. In the case of star topology, the Split CI, the CPI-EnKF and the Augmented EnKF produce practically equivalent state estimates. However, in the case of the path and the balanced tree topologies, both the CPI-EnKF and the Augmented EnKF outperform the Split CI algorithm as they produce estimates with a lower error. Another interesting result is the "ripples" in the average error of the CPI-EnKF estimates. These are caused by a numerical instability of the CPI-EnKF equations in the situation where no new observations are available

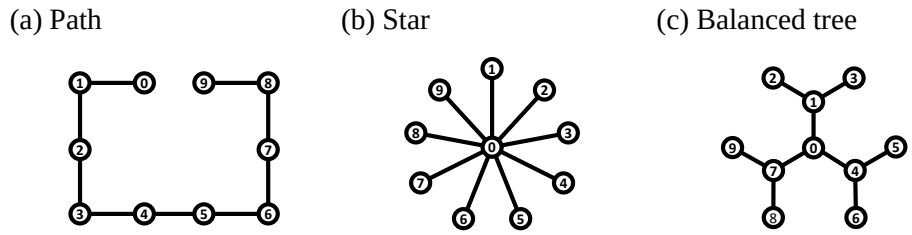


Fig. 6.11: Topologies of sensor networks evaluated.

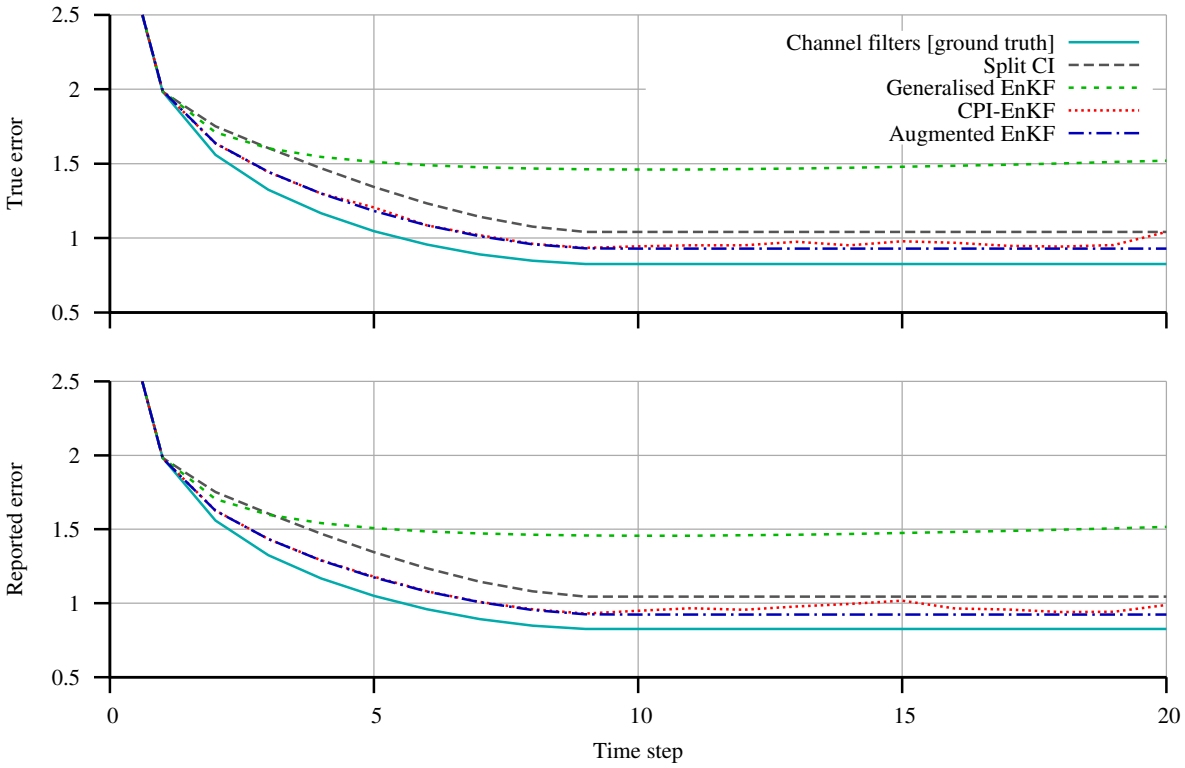


Fig. 6.12: Accuracy of EnKF variants in a sensor network with path topology.

in the system, as discussed in Section 5.2.2. The Augmented EnKF appears to be less sensitive to this problem, which makes it a better filter for applications where new observations are rare.

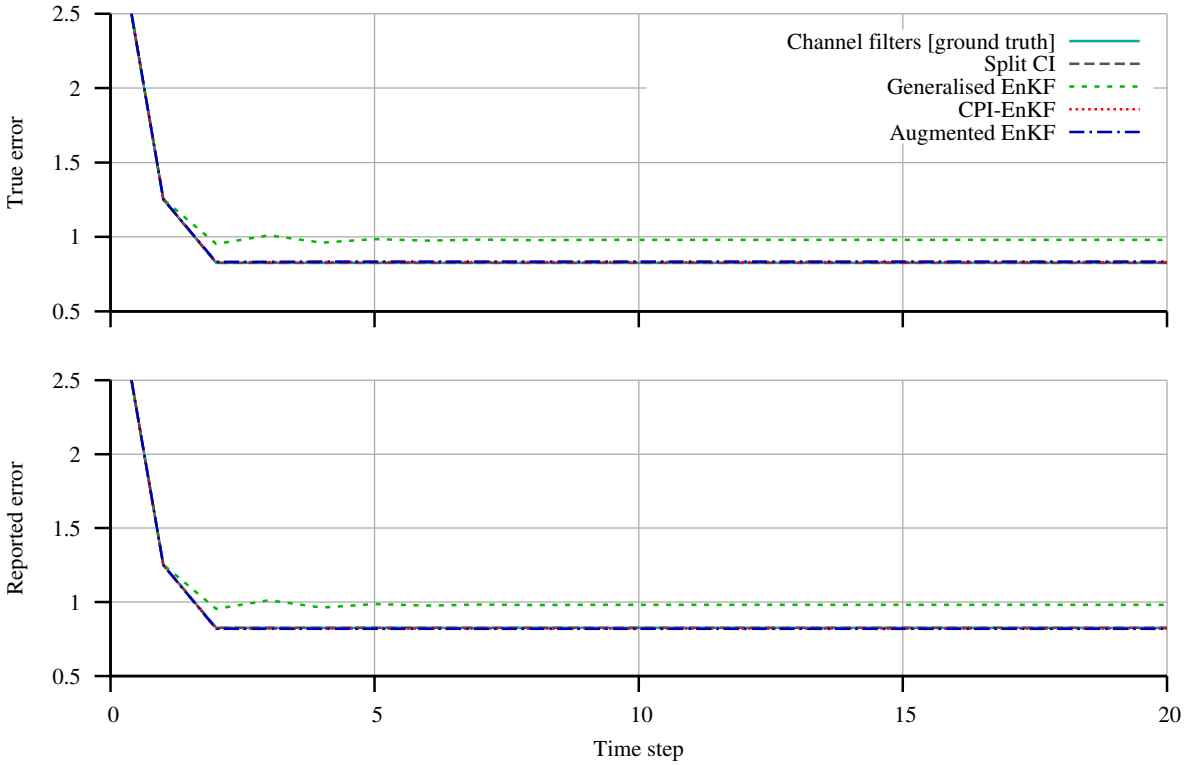


Fig. 6.13: Accuracy of EnKF variants in a sensor network with star topology.

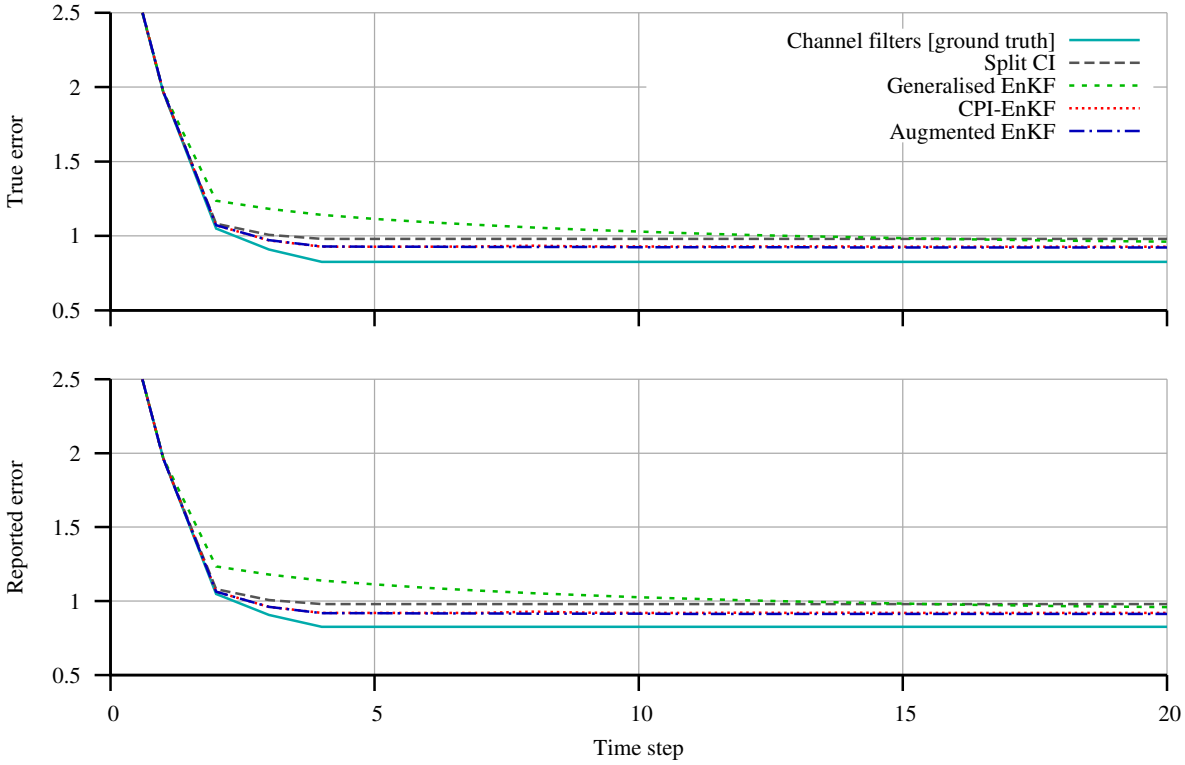


Fig. 6.14: Accuracy of EnKF variants in a sensor network with balanced-tree topology.

6.5 Chapter Summary

This chapter provided a comprehensive evaluation of the numerical properties of the new EnKF variants developed in this thesis in a wide range of correlated estimation problems. The principal insights obtained from this evaluation can be summarised in the following points:

- The conventional EnKF, even with noise compensation generated according to a correct model of correlation, is not a suitable filter for correlated estimation problems.

- The Generalised EnKF, which is a Monte Carlo equivalent of the Generalised Kalman filter, is not a good filter, and it is not suitable for correlated estimation problems.

- The experiments indicate that both the CPI-EnKF and the Augmented EnKF provide consistent state estimates in **all** the correlated estimation problems considered, which is a necessary condition for application of the filters to safety-critical systems.

- In estimation problems affected by correlated noises, neither the CPI-EnKF nor the Augmented EnKF achieve the accuracy of the algebraically-exact modifications of the Kalman filter. However, their use can be justified in certain types of such applications, for example, in those where state augmentation would incur an unacceptable performance penalty. One such application is presented in Section 7.3.

- In estimation problems affected by common past information, both the CPI-EnKF and the Augmented EnKF outperform the only feature-wise comparable algorithms - the CI and the Split CI, which makes both the new algorithms highly suitable for distributed data fusion problems. An example of one such application is provided in Section 7.1.

- The CPI-EnKF and the Augmented EnKF provide practically identical estimates in all the correlated estimation problems considered; although theoretically they can provide different estimates in certain circumstances, it is not clear which practical problems these circumstances map to.

- The Augmented EnKF is numerically more stable than the CPI-EnKF. Considering that the Augmented EnKF can also be used with non-square observation models $\mathbf{H}$ (see Section 5.2.3), the Augmented EnKF appears to be the most suitable variant of the EnKF applicable to correlated estimation problems.

- In correlated estimation problems, the covariance does not represent a full description of the situation, and optimal data fusion can only be performed under a valid interpretation of the origin of the correlation. The results provided by this thesis suggest that an universal update rule that would be optimal under all such interpretations might not exist at all, but no formal proof of such a statement is known to the author.

While the estimation problems provided in this chapter are rather theoretical, Chapter 7 presents several applications of the CPI-EnKF and the Augmented EnKF to practical real-world problems.

Chapter 7

Practical Evaluation

This chapter describes three applications of the CPI-EnKF and the Augmented EnKF filters to real-world problems in the larger field of robotic localisation. Each of the problems is introduced in the context of their respective areas, and a brief review of the appropriate state-of-the-art methods is provided. The work presented here not only serves as a guiding example of how the new filters can be applied to practical correlated estimation problems, but it also demonstrates that the EnKF in general has the merit in robotic applications and its exclusion from this field is rather unjustified.

7.1 Cooperative Localisation

This section considers the problem of cooperative vehicle localisation, in which a group of vehicles are driving in an outdoor environment, each estimating its position using a global positioning system (GPS) and odometry. Additionally, the vehicles can improve their estimates by observing positions of other vehicles using a proximity sensor, such as a radar, and mutual communication, which is especially helpful to those vehicles operating in areas with no GPS coverage.

Such a cooperative localisation problem is a prime example of a distributed data fusion problem in a sensor network. As discussed in Chapter 4, in order to achieve scalable data fusion in an arbitrary large network, the nodes (vehicles) need to communicate their estimates (positions) instead of raw sensor measurements, and therefore, the system will be exposed to the common past information problem, which is amplified by the dynamic and unstructured nature of the communication topology, inherent to a cooperative localisation scenario.

This section presents a new cooperative localisation system, which is based on the CPI-

EnKF filter developed in this thesis. It will be demonstrated that the new system is simpler to apply, provides better estimates, can be scaled to an arbitrary number of vehicles and is computationally more efficient than state-of-the-art cooperative localisation methods. The main goal of this section is to demonstrate the practical utility of the new methods developed in this thesis to address a real-world instance of the common past information problem. Note that the work presented here has been published in (Čurn et al., 2013).

7.1.1 Introduction

The past two decades witnessed a revolution in the deployment of advanced driver assistance systems. A significant number of vehicles currently available on public roads are equipped with GPS receivers and on-board navigation computers, which inform the driver about their current position on a map and advise them on navigation decisions. The built-in navigation computers often integrate vehicle odometry information obtained through wheel rotation counters or similar sensors, which helps the navigation computers to maintain the global position estimates even in areas with no GPS coverage, such as in urban canyons or tunnels. Although additional sensors, such as an inertial measurement unit (IMU), magnetic compass or visual odometry, can help the computer to maintain a more accurate global position estimate, without any global reference, the accuracy of the estimate always deteriorates as relative errors accumulate over time.

Various approaches to improve the accuracy of the global position estimates in the case of poor or no GPS signal reception have been proposed. For example, Google's self-driving cars employ highly detailed 3D maps of the environment and advanced on-board sensors in order to fix the global position of a vehicle in the map with a high accuracy (Thrun and Urmson, 2011). Another class of methods proposes extending the road infrastructure with active or passive beacons that broadcast their global position, so that nearby vehicles equipped with an appropriate sensor can improve their global position estimate (Lee et al., 2009). Unfortunately, all these approaches are costly, which impedes their wide-scale deployment to the public road network.

Cooperative localisation is based on a simple idea that vehicles with more accurate position estimates can help nearby vehicles improve their potentially poor position estimates. In technical terms, a vehicle can detect a relative location of a nearby vehicle using a proximity sensor such as a radar, a laser range finder (LIDAR) or a video camera. This information can then be communicated between the vehicles over a wireless network link and fused with their

current positions estimates, and consequently, improve the accuracy of the estimates. Such an improvement stems from the fact that the other vehicle might be driving through an area with better GPS coverage and thus have a better position estimate, and also because the fusion of observations from multiple independent sources generally leads to better estimates. It is important to note that cooperative localisation does not require any investments in the road infrastructure, while radars or video cameras are increasingly available in new cars, even in the mid-range segment, and a wireless communication capability is generally cheap. Furthermore, cooperative localisation can be seamlessly combined with other positioning systems and sensors.

7.1.2 Related Work

The state-of-the-art approaches to the problem of cooperative localisation can be divided into two principal classes, based on the type of information the vehicles communicate: either they only communicate local information obtained exclusively through a vehicle's own sensors, or they communicate information potentially obtained from other vehicles' sensors. Note that cooperative localisation is just another instance of a sensor network data fusion problem, and hence the methods described in the following sections are highly related or even based on the methods reviewed in Chapter 4.

In the first class of cooperative localisation approaches, each vehicle only communicates local (pre-processed) sensor measurements obtained exclusively using the vehicle's own sensors. Such measurements are typically statistically independent of other vehicles' measurements, and therefore the common past information problem is avoided. The measurements can be communicated to a single central authority that estimates the position of all the vehicles in the environment as one large system, using fusion algorithms such as the Extended Kalman filter (Mourikis and Roumeliotis, 2006; Huang et al., 2009; Roumeliotis and Rekleitis, 2004), particle filter (Fox et al., 2000) or Maximum Likelihood estimation (Howard et al., 2002). Although the centralised methods typically provide optimal global position estimates, they are susceptible to a single point of failure and are not scalable to a larger number of vehicles. In order to eliminate the single point of failure problem, the central filter can be decomposed into a set of communicating filters distributed among all the vehicles (i.e. decentralised), either in an optimal fashion, at the expense of a higher computation and communication cost (Roumeliotis and Bekey, 2002; Nerurkar et al., 2009), or in an approximate fashion, with more efficient communication and computation (Barooah et al., 2010). Another approach, described in (Leung et al., 2009), as-

sumes the vehicles only communicate sporadically and it proposes an algorithm that allows the vehicles to only store and communicate the smallest necessary set of sensor measurements that still guarantees optimal position estimation. Unfortunately, this is only possible for an a priori known and fixed number of vehicles. Alternatively, each vehicle can maintain an estimate of the state of all neighbouring vehicles based solely on the vehicle's own sensors, and broadcast this so-called *group state* to nearby vehicles, thus helping them to improve their own global position estimates (Karam et al., 2006); although the communication and computation is quite efficient in such a system, the information does not flow transitively between non-neighbouring vehicles and therefore, the position estimates might be of mediocre quality.

In the second class of approaches, the vehicles communicate data that is potentially inferred from other vehicles' sensor measurements - typically, the actual global position estimates. In principle, when a vehicle receives a global position estimate which was broadcast by a nearby vehicle, it combines it with information about the relative location of that vehicle obtained from an on-board sensor, and uses this combined information as an observation to improve it's own global position estimate, using a fusion algorithm such as the Kalman filter. Unfortunately, such an approach is susceptible to the common past information problem, because the global position estimate of the other vehicle might depend on the global position estimate of the local vehicle that had been broadcast earlier. In effect, this dependence causes a correlation between the local estimate error and the observation error and it violates the assumption of independence, inherent to the Kalman filter and many other data fusion algorithms. The state of the art offers various approaches to the common past information problem, such as to ignore it (Martinelli, 2007), to maintain an (inherently incomplete) dependency tree to limit the extent of so-called *circular updates* (Howard et al., 2003), or to use a sub-optimal but consistent algorithm such as Covariance Intersection (CI) to fuse the data (Li and Nashashibi, 2012).

Intuitively, in order to make a cooperative localisation system applicable to the existing public road network, the system must fulfill the following requirements. There can be an arbitrarily large and a priori unknown number of vehicles, which can join and leave the road network at any time. The vehicles can have different shapes, move erratically in the environment rather than in a particular formation, can employ different types of sensors, and they can be equipped with maps of varying precision, or no maps at all. Furthermore, wireless communication is generally not reliable and therefore, the cooperative localisation system must not depend on any well-defined communication pattern; instead, it should operate opportunistically on a best-effort basis whenever a communication channel between vehicles can be established. The next section

proposes a cooperative localisation system based on the CPI-EnKF filter, which meets all the above requirements.

7.1.3 Approach

Assume each vehicle maintains an individual Ensemble Kalman filter (EnKF), as described in Section 2.6, in order to estimate its own global position using data from on-board sensors, such as the GPS, odometry, or any other suitable sensor. The design and implementation of such an estimation system is a standard task, as mathematical models for many vehicle and sensor types are well understood and generally available. Additionally, the vehicles are equipped with one or more proximity sensors, such as a radar or video camera, that enable them to detect relative locations of nearby vehicles. When a vehicle detects another vehicle in its vicinity, it can establish a vehicle-to-vehicle (V2V) communication channel, enabling the two to exchange their current global position estimates and measured relative displacement, and use that information to update their local position estimates. The following paragraph describes how to perform such an update using the CPI-EnKF.

Formally, at a discrete time step $t-1$, a vehicle represents the current state estimate using an ensemble $\mathbf{X}_{t-1} \in \mathbb{R}^{n \times N}$. The state is composed of variables that describe its global position in the world (e.g. latitude, longitude, direction), as well as other variables potentially needed to model the vehicle dynamics (e.g. steering angle, speed, acceleration). The cooperative localisation algorithm repeats the following steps:

1. Given the vehicle's last state estimate represented as an ensemble $\mathbf{X}_{t-1}$, predict the state at the next discrete time step t using Equation (2.84), with a prediction model based either on ego-motion sensors (e.g. odometry) or a vehicle dynamics model. The resulting state estimate is represented as a *prior ensemble* $\mathbf{X}_t^- \in \mathbb{R}^{n \times N}$.

2. Update the prior ensemble $\mathbf{X}_t^-$ using Equations (2.86) and (2.87) in order to assimilate the current positional sensors measurements (e.g. GPS, IMU, compass), and thus obtain a more accurate state estimate represented as a *communication ensemble* $\mathbf{X}_t^* \in \mathbb{R}^{n \times N}$. Such sensor measurements can be assumed to be affected by a white Gaussian noise, and therefore, the traditional Kalman gain formula can be applied.

3. Measure the relative displacement $\mathbf{r}_t \in \mathbb{R}^m$ of a neighbour vehicle using a proximity sensor (e.g. radar, LIDAR, video camera). Let $\mathbf{R}_t \in \mathbb{R}^{m \times m}$ denote a covariance matrix charac-

terising the error of the measurement.

4. Send a message to the neighbour vehicle containing the relative displacement $\mathbf{r}_t$, covariance $\mathbf{R}_t$, and the communication ensemble $\mathbf{X}_t^*$ (or only a subset thereof limited to the global position-related state variables).

5. Receive a message submitted by the neighbour vehicle containing its own communication ensemble, herein denoted as $\mathbf{Y}_t \in \mathbb{R}^{n \times N}$.

6. Update the prior ensemble $\mathbf{X}_t^-$ using Equations (5.42) and (5.43) to accommodate the "observation" indicating the vehicle's location, which is an ensemble constructed from the received ensemble $\mathbf{Y}_t$, the relative displacement measurement $\mathbf{r}_t$ and randomly generated samples with covariance $\mathbf{R}_t$. The CPI-EnKF formula needs to be applied in this step because $\mathbf{X}_t^-$ and $\mathbf{Y}_t$ are potentially correlated as they might share common past information. The result of such an update leads to a more accurate state estimate represented by a *provisional posterior ensemble* $\mathbf{X}_t^+ \in \mathbb{R}^{n \times N}$.

7. Update the provisional posterior ensemble $\mathbf{X}_t^+$ using Equations (2.86) and (2.87) to accommodate the current positional sensors measurements (similarly to Step 2), leading to the *final posterior ensemble* $\mathbf{X}_t$ that represents the current best state estimate.

In principle, this opportunistic distributed cooperative localisation algorithm is very similar to the decentralised Covariance Intersection data fusion algorithm introduced in (Julier and Uhlmann, 2001a), which is reviewed in Section 4.6, the main difference being that the above algorithm operates with ensembles instead of means and covariances. The efficiency of the algorithm stems from that fact that it exploits the locality in the communication at the expense of sub-optimality of the position estimates compared to a theoretical central estimator - the sensor information is only propagated locally between neighbouring vehicles and therefore, some vehicles will not receive it and will not update their position estimates.

Note that the algorithm assumes that the ensembles maintained by all the vehicles have the same number of samples. Although the algorithm is described from the perspective of an observing vehicle, the steps taken by the vehicle being observed are very similar because it practically makes no difference which of the two vehicles performs the actual relative displacement measurement and initiates the communication. The steps of the algorithm are idealised in the sense they assume a single neighbour vehicle is detected at every time step; in practice, if two or more vehicle were detected, Steps 3-6 need to be repeated separately for each of the vehicles

detected, or not performed at all if no vehicle was detected. Furthermore, it is assumed that all the vehicles have some unspecified means to associate an observed neighbour vehicle to the vehicle communicated to, and also that all the vehicles operate at synchronous time steps. In practice, additional provisions need to be made to ensure correct data association and timing.

In order to evaluate the accuracy of the CPI-EnKF cooperative localisation algorithm proposed in this section, its position estimates will be compared with estimates provided by the following algorithms: the Augmented Ensemble Kalman filter (Augmented EnKF), the Covariance Intersection (CI), the Split Covariance Intersection (Split CI), and the local Kalman filter (Local KF). These algorithms, all of whom will be described in detail in the following sections, have been chosen for this evaluation because they all meet the criteria outlined in the beginning of Section 7.1: they are fully decentralised, cost a constant space and time per update, can be scaled up to an arbitrary number of vehicles, support diverse vehicle types and sensors, only assume opportunistic communication and, apart from the Local KF algorithm, they all provide consistent estimates, given certain assumptions. The Local KF algorithm is only included to illustrate the problem of overconfidence and divergence should the common past information be neglected. Additionally, the evaluation also includes a centralised Kalman filter (Central KF) algorithm to show the optimal position estimates as a reference.

7.1.4 Scenario

The world in the evaluation scenario is best imagined as a two-dimensional city consisting of 16 rectangular city blocks, organised in a 4-by-4 grid, with each block of dimension 150-by-100 m, as depicted in Figure 7.1. There are in total 16 vehicles driving around the blocks on "roads" that are, for simplicity of implementation, assumed to be straight lines of zero width. The vehicles, which themselves are modelled as simple points, only drive on these straight lines and collisions between the vehicles are not considered. The speed of the vehicles varies from 1 to 20 m/s, and it changes with an acceleration that evolves randomly over time as a Wiener process bounded between -4 and 4 m/s^2, rescaled to have an absolute standard deviation of 0.632 m/s^2 per second. Whenever a vehicle reaches the end of a city block, it randomly decides whether to take a turn (not a U-turn though) or to continue. This action affects neither the vehicle's speed nor its acceleration. The evolution of the world is simulated in discrete time steps of 0.1 s. The vehicles start off with an initial global position estimate that is consistent with their true position and has an error with a standard deviation of 10 m.

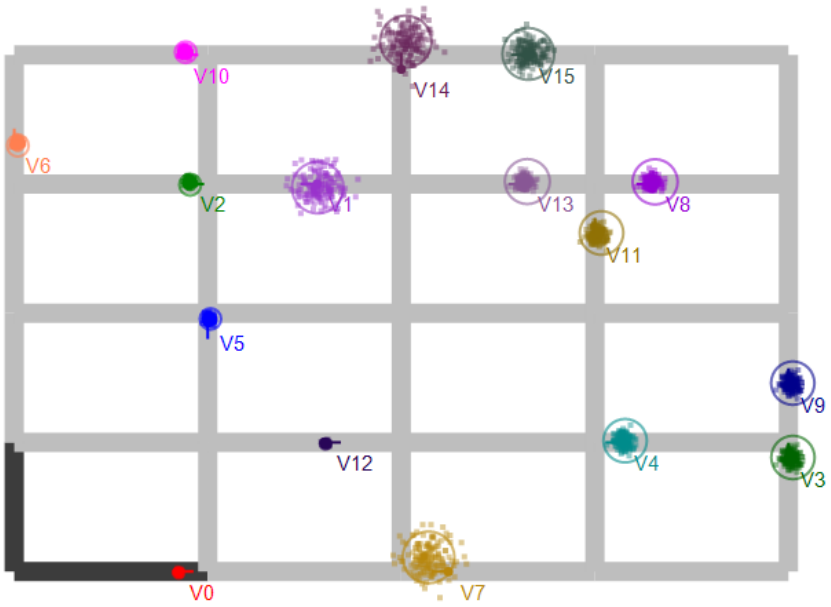


Fig. 7.1: Road configuration in the cooperative localisation evaluation scenario, including an example snapshot of the vehicles' position estimates from a simulation run. The darker segment in the bottom left section represents the only road with GPS coverage. The points depict the current vehicle position estimates (ensembles) of the respective CPI-EnKF filters, while the solid ellipses depict the 95% confidence regions of the (apparently more conservative) Split CI estimates.

Each vehicle is equipped with a (simulated) odometry sensor, that reports a relative two-dimensional displacement of the vehicle since the previous reading, affected independently in both axes by a zero-mean white Gaussian noise with a standard deviation of 0.05 m for every meter of the true distance travelled. The odometry has no angular error, in order to avoid a bias in the evaluation results caused by non-linear effects. Each vehicle is also equipped with a GPS receiver, that enables them to detect their global position, affected by a zero-mean white Gaussian noise with a standard deviation depending on the vehicle's location in the city. The odometry sensor provides readings with a period of 0.1 s (i.e. every simulation time step), while the GPS receiver reports the global position with a period of 1 s (i.e. every 10th simulation time step).

Additionally, each vehicle is equipped with a proximity sensor that can detect the relative distance in two dimensions to nearby vehicles driving on the same straight road segment, with a probability of making an actual observation (at any given simulation time step) that decreases linearly from 15% to 0% as the distance to the other vehicle increases from 5 m to 100 m. The proximity sensor readings are affected independently in both axes by a zero-mean white

Gaussian noise with a standard deviation of 0.05 m for every meter of the true distance between the vehicles. One vehicle can observe multiple vehicles during a single time step, and it can always establish a communication channel with all the vehicles observed. The communication is instant. Note that it would be counterproductive to perform the cooperative localisation data exchanges too often, because the chance that subsequent information exchanges in a close group of vehicles bring any new information is quite low, while these exchanges still contribute to the accumulation of numerical errors, and in real-world applications they congest the wireless communication medium.

As argued in Section 7.1.1, cooperative localisation has the most significant impact on vehicles that have very inaccurate global position estimates, such as vehicles operating in areas with a poor GPS coverage. In order to introduce this effect in this evaluation scenario, a GPS signal is only available on one road segment of the city (see the darker bottom left region in Figure 7.1). In this area, the GPS signal has such a quality that enables the receiver to compute the global position with an error equivalent to a zero-mean white Gaussian noise with a standard deviation of 3 m. All the other areas in the city have no GPS coverage at all, and therefore, for vehicles driving in these areas the cooperative localisation represents the only available means to maintain position estimates with a reasonably accuracy.

Although the evaluation scenario involves a simplistic model of the world, it suffices as a tool to compare the various cooperative localisation algorithms. Importantly, the simulated world meets all the theoretical assumptions of the algorithms being evaluated, such as that the system dynamics are linear and all the errors involved are Gaussian, and therefore, all the algorithms have perfect conditions for their operation. Also, note that all the constants in the presented scenario are chosen in order to model real-world city traffic as closely as possible, but similar results can also be achieved with a different constant selection.

7.1.5 Implementation

For the purpose of this evaluation, several variants of the opportunistic distributed cooperative localisation algorithm described in Section 7.1.3 have been implemented. These variants, which will be discussed in this section, differ mainly in the way they represent the state estimates and how they accommodate the potentially correlated observations. All of them model the vehicle's system state as a two-dimensional vector that represents the absolute coordinates of the vehicle in the city grid, measured in meters, with the coordinate origin in the bottom-left corner of the

map.

All the evaluated variants of the cooperative localisation algorithm facilitate the prediction in Step 1 simply by altering the state estimate using a two-dimensional odometry reading and an associated covariance matrix, and the update in Steps 2 and 7 using a two-dimensional GPS reading and an associated covariance matrix, whenever available. The relative displacement of a neighbour vehicle measured in Step 3 is represented as a two-dimensional vector, i.e. $\mathbf{r}_t \in \mathbb{R}^2$ and $\mathbf{R}_t \in \mathbb{R}^{2\times 2}$. Steps 4 and 5 do not require any implementation, as all the necessary information is implicitly available to the simulation process. Therefore, the following text will only focus on the description of the implementation of Step 6 by the particular algorithms, which is arguably the only non-trivial part.

In the implementation of the base CPI-EnKF variant of the cooperative localisation algorithm, the current state estimate is represented using an ensemble with the number of samples $N = 1000$. As described in Step 6, the provisional posterior ensemble $\mathbf{X}_t^+ \in \mathbb{R}^{2\times 1000}$ is computed using Equations (5.42) and (5.43). The inputs to these equations are defined as follows:

$$\mathbf{X} = \mathbf{X}_t^- \tag{7.1}$$

$$\mathbf{Z} = \mathbf{Y}_t - \mathbf{V}_t \tag{7.2}$$

where $\mathbf{X} \in \mathbb{R}^{2\times 1000}$ denotes the state estimate, $\mathbf{Z} \in \mathbb{R}^{2\times 1000}$ denotes the observation, and $\mathbf{V}_t = \left[\mathbf{v}_t^1, \ldots, \mathbf{v}_t^{1000}\right] \in \mathbb{R}^{2\times 1000}$ is a Gaussian observation noise compensation term whose columns are randomly generated as:

$$\mathbf{v}_t^i \leftarrow \mathcal{N}(\mathbf{r}_t, \mathbf{R}_t) \tag{7.3}$$

for $i = 1, \ldots, 1000$. The observation model $h\colon \mathbb{R}^2 \to \mathbb{R}^2$ is simply an identity function, and therefore:

$$h(\mathbf{X}) = \mathbf{X}_t^- \tag{7.4}$$

Recall that $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a (multivariate) Gaussian distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$.

The Augmented EnKF variant of the algorithm equivalent to the CPI-EnKF variant described above, the only difference is that the Augmented EnKF update rule, as given in Equations (5.61) and (5.62), is used instead of the CPI-EnKF update rule.

In the CI variant of the cooperative localisation algorithm, the state estimate is represented using a two-dimensional mean vector and a covariance matrix. These values are communicated

between vehicles instead of the ensembles. As such, the whole algorithm is de facto equivalent to the distributed data fusion algorithm described in (Julier and Uhlmann, 2001a), and also reviewed in Section 4.6.

Suppose the prior state estimate is denoted $\hat{\mathbf{x}} \in \mathbb{R}^2$ and the associated error covariance $\mathcal{X} \in \mathbb{R}^{2\times 2}$, and the state estimate received from the other vehicle is denoted $\hat{\mathbf{y}} \in \mathbb{R}^2$ and the associated error covariance $\mathcal{Y} \in \mathbb{R}^{2\times 2}$. Then in Step 6, the provisional posterior estimate $\hat{\mathbf{x}}^+ \in \mathbb{R}^2$ and the associated error covariance $\mathcal{X}^+ \in \mathbb{R}^{2\times 2}$ is computed using the CI update rule (see also Definition 4.2) as follows:

$$\mathcal{X}^+ = \left[\omega\mathcal{X}^{-1} + (1-\omega)\mathcal{Y}^{-1}\right]^{-1} \tag{7.5}$$

$$\hat{\mathbf{x}}^+ = \mathcal{X}^+\left(\omega\mathcal{X}^{-1}\hat{\mathbf{x}} + (1-\omega)\mathcal{Y}^{-1}\hat{\mathbf{y}}\right) \tag{7.6}$$

with the coefficient $\omega \in \mathbb{R}$ optimised on the fly so that the trace of the covariance matrix $\mathcal{X}^+$ is minimal. Note that the CI update rule is immune to the correlation between $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, and therefore, it provides a consistent estimate in this application (Julier and Uhlmann, 1997a).

In the Split CI variant of the cooperative localisation algorithm, the state estimate is also represented using a two-dimensional mean vector. However, as opposed to the CI, the error covariance matrix is split into two additive component covariance matrices: the first covariance matrix represents the potentially correlated error component, and the second covariance matrix represents the known-independent error component. Such a splitting of the covariance leads to a higher accuracy of the estimation, compared to the CI (Julier and Uhlmann, 2001a).

Due to the fact that the vehicle's on-board sensors produce independent observations, the prediction step using the odometry readings will only affect the known-independent error component of the state estimate, and the update using the GPS readings will be optimal because the correlated component of the GPS error is zero. Note that such an algorithm is similar to the cooperative localisation algorithm presented in (Li and Nashashibi, 2012).

Again, denote the prior state estimate and the state estimate received from the other vehicle as $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, respectively, and the corresponding error covariance matrix components as $\mathcal{X}_1 + \mathcal{X}_2$ and $\mathcal{Y}_1 + \mathcal{Y}_2$, respectively. Step 6 of the cooperative localisation algorithm computes the provisional posterior estimate $\hat{\mathbf{x}}^+$ and the associated covariance matrix components $\mathcal{X}_1^+ + \mathcal{X}_2^+$

using the Split CI update rule (see also Definition 4.3) as follows:

$$\mathcal{X}^{-1} = \omega \left(\mathcal{X}_1 + \omega \mathcal{X}_2\right)^{-1} \tag{7.7}$$

$$\mathcal{Y}^{-1} = (1-\omega) \left(\mathcal{Y}_1 + (1-\omega) \mathcal{Y}_2\right)^{-1} \tag{7.8}$$

$$\mathcal{X}_1^+ = \left(\mathcal{X}^{-1} + \mathcal{Y}^{-1}\right)^{-1} \tag{7.9}$$

$$\mathcal{X}_2^+ = 0 \tag{7.10}$$

$$\hat{\mathbf{x}}^+ = \hat{\mathbf{x}} + \mathcal{X}_1^+ \mathcal{Y}^{-1} \left(\hat{\mathbf{y}} - \hat{\mathbf{x}}\right) \tag{7.11}$$

Again, $\omega \in \mathbb{R}$ is optimised on the fly so that the trace of the covariance matrix $\mathcal{X}_1^+$ is minimal. Note that the above equations differ slightly from the standard Split CI update rule provided in Definition 4.3, because the known-independent covariance component is directly added to the correlated component. This is necessary to ensure a consistency of the cooperative localisation algorithm, because the resulting position estimate might be correlated fully to subsequent position estimates received from other vehicles via the communication.

The Local KF variant of the cooperative localisation algorithm represents, similarly to the CI, the state estimate using a mean vector and a covariance matrix, and the update in Step 6 is performed using a simple Kalman filter update rule, as described in Definition 2.4. Assuming the same notation as in the CI variant above, the provisional posterior state estimate is computed as follows:

$$\mathbf{K} = \mathcal{X}[\mathcal{X} + \mathcal{Y}]^{-1} \tag{7.12}$$

$$\mathcal{X}^+ = \mathcal{X} - \mathbf{K}\mathcal{X} \tag{7.13}$$

$$\hat{\mathbf{x}}^+ = \hat{\mathbf{x}} + \mathbf{K}(\hat{\mathbf{y}} - \hat{\mathbf{x}}) \tag{7.14}$$

As discussed in Section 2.1, the Kalman filter update rule assumes the prior state estimate $\hat{\mathbf{x}}$ is independent of the observation $\hat{\mathbf{y}}$, which in this case does not hold. Therefore, the resulting provisional posterior state estimate $\hat{\mathbf{x}}^+$ with the error covariance $\mathcal{X}^+$ might be inconsistent with the true error, the filter might become overconfident over time and diverge. The Local KF algorithm is included merely to illustrate the consequences of this problem.

Finally, the position estimates of all the vehicles in the system are also computed using the Central KF algorithm, which has been described in Section 4.1. The system state vector includes the positions of all the vehicles in the simulation. In such a model, all sensor observations are independent of the state. Because all the Kalman filter assumptions are satisfied, the resulting state estimates are guaranteed to be consistent and optimal, given all the available sensor observations (Simon, 2006). Therefore, the Central KF represents an upper limit of the quality of

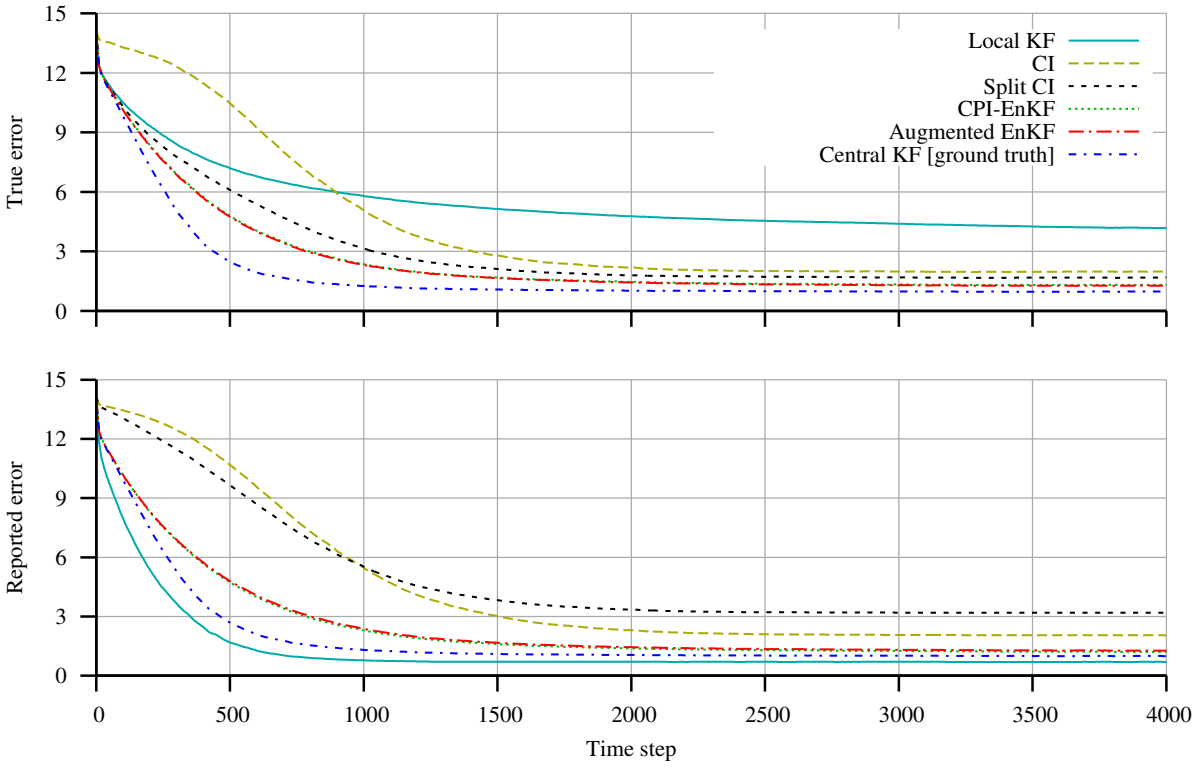


Fig. 7.2: Accuracy of position estimates provided by various cooperative localisation algorithms, averaged over 1000 random simulations. The first graph shows an average root mean square (RMS) position error of the estimates (lower value means smaller errors). The second graph shows an average variance of the errors reported by each of the estimation algorithms (for consistent algorithms, this value should be greater than or equal to the average true position error in the first graph; the closer the better)

the other cooperative localisation algorithms and as such it serves as a useful reference in the evaluation.

7.1.6 Results

The system simulation has been executed 1000 times with a different random seed for every run, and the statistics of the position estimates provided by all the filters in every time step were aggregated over all the simulation runs, in a similar way as the evaluation in Chapter 6 was performed. Figure 7.1 depicts a sample snapshot of one of the simulation runs, and Figure 7.2 shows the graphs illustrating the quality of position estimates provided by the particular cooperative localisation algorithms, averaged over all the simulation runs. The most important results from this evaluation are the following:

- In every aspect evaluated, the CPI-EnKF algorithm is superior to the CI, the Split CI and the Local KF algorithms.

- As expected in this particular scenario, the Augmented EnKF and the CPI-EnKF provide de facto equivalent estimates.

- The evaluation confirms that the Local KF is inconsistent because it neglects the correlations.

- While the Split CI algorithm provides reasonable position estimates, the associated covariances are overly pessimistic - see also Figure 7.1.

- After a long enough time, all the algorithms reach an equilibrium where position errors stemming from the inaccurate odometry are, on average, compensated for by the gains in the accuracy due to the GPS and the cooperative localisation. The average magnitude of the position errors in this equilibrium depends on the quality of the cooperative localisation algorithm.

7.1.7 Conclusion

This section discussed the problem of cooperative localisation in a group of communicating vehicles, described the state-of-the-art approaches that address it, and then developed an entirely new approach based on the CPI-EnKF filter introduced in this thesis. It has been argued that the new approach, unlike the majority of the existing ones, provides consistent estimates given certain assumptions, it is simple to tune and implement, supports arbitrary vehicle types and sensors, and it can be scaled to an unlimited number of vehicles. The performance of the new CPI-EnKF cooperative localisation algorithm has been compared to existing similar algorithms in a comprehensive evaluation.

In order to apply the new cooperative localisation algorithm in practice, several issues need to be resolved, such as a correct association between the vehicles observed and the vehicles communicated with, optimal planning of the information exchanges and an associated efficient allocation of the communication bandwidth, in particular in areas with a large number of vehicles.

7.2 Simultaneous Localisation and Mapping

This section deals with simultaneous localisation and mapping (SLAM), a problem in which a mobile robot operating in a previously unknown environment is building a map of that environment, while simultaneously using the map for navigation. As shown in (Smith et al., 1990), SLAM can be viewed as an estimation problem - the goal of the robot is to estimate its own position and the position of objects in its environment from a sequence of noisy sensor observations. Although SLAM is not a correlated estimation problem per se, this section develops a method which transforms the SLAM problem to a problem of data fusion in the presence of common past information, discussed in Chapter 4, and applies the new CPI-EnKF filter to address it. The main goal of this section is to demonstrate the versatility of the EnKF-based correlated estimation methods developed in this thesis, which makes their application useful even in estimation problems not affected by correlations, by offering some novel features with respect to existing state-of-the-art methods in corresponding areas.

More specifically, in the approach presented in this section, the uncertain spatial constraints between features in a robot's environment are represented as ordered sets of Monte Carlo samples drawn from the space of coordinate frame transformations. Such a representation enables fusion of two or more spatial constraints using the update rule of the CPI-EnKF even if their errors are correlated. The spatial constraints are organised in a compact data structure, which models the full posterior over the robot's pose and landmark locations. The number of Monte Carlo samples necessary to accurately represent the posterior does not grow exponentially with the number of state-space dimensions as in conventional particle filters - in fact, it is a constant parameter. This data structure provides a constant time access to marginal distributions and a newly observed spatial constraint can be accommodated in time linear to the number of landmarks tracked, regardless of the number of spatial constraints that have been observed previously. The section provides an experimental evaluation of the new approach, and a discussion of its strengths and weaknesses with respect to the well-established SLAM approaches. The work presented here has been published in (Čurn et al., 2012a).

7.2.1 Related Work

The problem of SLAM has attracted a significant amount of research in the past two decades. Historically, the most popular approaches are built around three major paradigms: Extended Kalman filters (EKF), particle filters and graph-based optimisers. The earliest solution to the

SLAM problem was published in a seminal paper by Smith, Cheeseman and Self (Smith et al., 1990). In their work, the current best estimate of the pose of a robot and features in its environment is represented using a high-dimensional vector, and the uncertainty of the estimate using a covariance matrix of the same dimension. In every step, the state is updated with new measurements using the EKF. Subsequently, this method became known as EKF-SLAM. A large number of EKF-SLAM variants have been successfully applied to a wide range of robotic applications, including ground vehicles, aircraft, and underwater vehicles (Thrun et al., 2005). The principal limitation of EKF-SLAM is the fact that the state covariance matrix grows quadratically with the number of spatial features tracked, as does the time complexity per update. A number of methods, such as (Leonard and Jacob, 1999), (Guivant and Nebot, 2001) or (Paz et al., 2007), address this issue by decomposing the map into smaller submaps whose covariances are maintained separately. Another important limitation of the EKF-SLAM approaches, which limits the range of practical applications, are the strong assumptions on the measurement noise, in particular that it must be Gaussian and white (i.e. uncorrelated). The Monte Carlo stochastic map presented here only requires a space and time per update linear to the number of spatial features, without the need to selectively discard cross-correlations between features. The accuracy of the cross-correlations only depends on the number of Monte Carlo samples used to represent the uncertain spatial constraints, which is a constant parameter. Furthermore, the stochastic map preserves spatial constraints with arbitrary (non-Gaussian) distributions, until a newly observed spatial constraint closes a loop.

The second major paradigm for addressing the SLAM problem is that based on particle filters (Doucet and Johansen, 2009). Its general idea is to represent the posterior by a set of particles, where each particle captures one particular sample of the pose of the robot or features in its environment. The posterior probability distribution is modelled with arbitrary precision in the limit, as the number of particles grows to infinity. After every observation, the particles are updated to form a proposal distribution, and sequential importance resampling selects a subset of the particles to model the target distribution - the new posterior. The main limitation of such a naive approach is that the number of particles necessary to accurately represent the posterior probability distribution grows exponentially with the number of dimensions, i.e. with the number of landmarks in the map. A number of methods, such as (Grisetti et al., 2005), (Eliazar and Parr, 2003), or more prominently FastSLAM (Montemerlo and Thrun, 2007), address the problem of exponential growth in the number of particles by a method called Rao-Blackwellisation. For example, the FastSLAM algorithm builds on an assumption that cross-correlations between

landmarks are independent given the robot's path, and each particle stores a robot's pose and a list of mean/covariance pairs of the landmark locations. The principal issue of such methods is the fact that the resampling step discards low-probability particles, and duplicates the high-probability ones. This means that the correlation information between landmarks is gradually being lost over time, which can lead to underestimation of the covariance and may cause problems when closing large loops. Similarly to the particle filtering-based methods, the approach presented in this section also represents the posterior by a set of Monte Carlo samples of the location of spatial features. However, the samples retain a certain order, which is necessary to capture cross-correlations between the features. Moreover, as sequential importance resampling is not used, the number of samples necessary for an accurate representation is independent of the number of spatial features tracked, and the cross-correlation information is not lost over time.

The third major SLAM paradigm, and arguably the most popular one in recent years, is based on graph-based non-linear optimisation. The basic idea, first formulated by Lu and Milios (Lu and Milios, 1997), is that all of a robot's poses and landmarks at a particular time represent nodes of a graph, and the spatial constraints between the poses represent the edges. The goal of an optimisation algorithm is to find a spatial configuration of the nodes that is most consistent with the constraints provided by the edges. Since the original Lu and Milios formulation, a large number of optimisation methods have been proposed, including iterative methods such as (Olson et al., 2006; Ranganathan et al., 2007), or direct solvers (Dellaert and Kaess, 2006; Frese, 2006). The state-of-the-art algorithms take advantage of the domain knowledge and sparsity of the constraint graph to perform the optimisation efficiently, provide efficient access to marginal covariances of observed features necessary for data association, and their implementations are readily available in frameworks such as iSAM (Kaess et al., 2008) and g^{2}o (Kummerle et al., 2011). The Monte Carlo stochastic map also maintains the spatial constraints in a graph structure. However, this graph only has a star topology; the spatial constraints between leaf nodes of the star are not represented explicitly in the graph, but implicitly by the ordered Monte Carlo samples saved in the nodes. Unlike graph-based optimisers, the performance of the stochastic map does not depend on the number of spatial constraints observed. Moreover, tracked spatial features can be discarded without affecting the cross-correlations between preserved features, in order to maintain real-time performance in on-line perception systems.

The SLAM approach presented in this section has certain similarities with all three major SLAM paradigms. The posterior over a robot's pose and landmark locations is modelled using a

set of Monte Carlo samples, however, unlike in particle filter-based methods, with a given order of samples, which is used to capture cross-correlations between spatial features. The samples are organised in a star-graph structure, where robot poses and landmarks represent the nodes, and uncertain spatial constraints between them represent the edges. Finally, the probability distributions modelled by the ordered Monte Carlo sample sets are merged using the CPI-EnKF algorithm developed in this thesis.

7.2.2 Representation of Spatial Constraints

The approach presented in this section assumes that a spatial constraint between two features can be represented as a probability distribution over the space of coordinate transformations between the two corresponding coordinate frames, i.e. an approximate transformation (AT). Only three-dimensional space will be considered, and a quaternion parametrisation of orientations (Altmann, 1986) adopted. However, a similar approach can also be applied to a two-dimensional space, and other parametrisations of coordinate transformations. The coordinate transformation vector (x, y, z, q) has seven variables, where q is a quaternion $q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ with components $a, b, c, d \in \mathbb{R}$ and imaginary units $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$.

An AT with a general probability distribution can be represented using a number of random Monte Carlo samples drawn from the space of all transformation vectors between two coordinate frames. Let $t \to u$ denote the set of all ATs from a frame t to a frame u. An AT $\mathbf{A} \in t \to u$ is represented as an ordered set of samples as:

$$\mathbf{A} \cong \left\{ (x_A^i,\, y_A^i,\, z_A^i,\, q_A^i) \mid i = 1, \ldots, N \right\} \tag{7.15}$$

where N is the number of samples, and each $(x_A^i,\, y_A^i,\, z_A^i,\, q_A^i)$ represents a single coordinate transformation vector. Many parametric probability distributions can easily be converted to such a representation by random sampling. For brevity, in the following text the index will be omitted from the notation of the samples, and also the following simplified syntax will be used instead of Equation (7.15):

$$\mathbf{A} \cong (x_A,\, y_A,\, z_A,\, q_A) \tag{7.16}$$

whenever the context will allow. Three elementary operations can be performed on the ATs: *compounding*, *inversion* and *merging*. All of these operations will be described in detail in the following sub-sections. Note that compounding and inversion are trivial vector operations, and are only described for the sake of completeness.

7.2.2.1 Compounding

If the robot has observed two consecutive ATs $\mathbf{A} \in t \to u$ and $\mathbf{B} \in u \to v$, such as $\mathbf{A} \cong (x_A,\, y_A,\, z_A,\, q_A)$ and $\mathbf{B} \cong (x_B,\, y_B,\, z_B,\, q_B)$, respectively, then the estimate of AT $t \to v$ can be computed using the *compounding* operation $\oplus$ as:

$$\mathbf{A} \oplus \mathbf{B} \cong (x_A + x,\, y_A + y,\, z_A + z,\, q) \tag{7.17}$$

where

$$q \;=\; q_A \times q_B \tag{7.18}$$

$$0 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \;=\; q_A \times (0 + x_B\mathbf{i} + y_B\mathbf{j} + z_B\mathbf{k}) \times q_A^{-1} \tag{7.19}$$

The $\times$ operation denotes quaternion multiplication and $^{-1}$ quaternion inversion; the quaternions q_A and q_B are assumed to be normalised, i.e. have unit length. More details on quaternions and spatial rotations can be found in (Altmann, 1986).

In other words, each sample of a compound AT can be acquired by combining the two samples at the same index from the input ATs. A sequence of ATs can be compounded by successively compounding pairs of neighbouring ATs; the order of ATs is irrelevant, because compounding is an associative operation:

$$(\mathbf{A} \oplus \mathbf{B}) \oplus \mathbf{C} = \mathbf{A} \oplus (\mathbf{B} \oplus \mathbf{C}) \tag{7.20}$$

Note that if two ATs are independent, a random permutation of the samples in one of the ordered sets would not affect the estimate of the compound AT. However, if the ATs are not independent, the order of samples needs to be retained, in order to provide an accurate estimate.

7.2.2.2 Inversion

Suppose $\mathbf{A} \in t \to u$ is an AT given as $\mathbf{A} \cong (x_A,\, y_A,\, z_A,\, q_A)$. The *inversion* operation $^{\mathbf{R}}$ computes the estimate of the AT in the opposite direction, $\mathbf{A}^{\mathbf{R}} \in u \to t$, as follows:

$$\mathbf{A}^{\mathbf{R}} \;\cong\; (x,\, y,\, z,\, q_A^{-1}) \tag{7.21}$$

where:

$$0 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \;=\; q_A^{-1} \times (0 - x_A\mathbf{i} - y_A\mathbf{j} - z_A\mathbf{k}) \tag{7.22}$$

Again, q_A must be a normalised quaternion. The inversion operation simply reverses a coordinate transformation of each sample from the input AT. Note that $^{\mathbf{R}}$ is a unary operator and an involution, i.e. $\left(\mathbf{A}^{\mathbf{R}}\right)^{\mathbf{R}} = \mathbf{A}$.

7.2.2.3 Merging

If the robot observes a spatial constraint between two objects multiple times, it can use all the observations to refine its knowledge about that relationship. In terms of approximate transformations, it is possible to combine two parallel ATs in order to produce a single AT with an uncertainty lower than the uncertainty of both original ATs. This operation corresponds to the multiplication of the probability distributions of the ATs, and it is derived from the conventional Ensemble Kalman Filter (EnKF), which has been reviewed in Section 2.6.

Consider two uncorrelated ATs $\mathbf{A} \in t \to u$ and $\mathbf{B} \in t \to u$. If quaternions are expanded to their four real components, samples of ATs $\mathbf{A}$ and $\mathbf{B}$ can be considered as seven-dimensional vectors $\mathbf{a}^1, \ldots, \mathbf{a}^N \in \mathbb{R}^7$ and $\mathbf{b}^1, \ldots, \mathbf{b}^N \in \mathbb{R}^7$, respectively, which can be organised into ensembles $\mathbf{P_A}$ and $\mathbf{P_B}$ as:

$$\mathbf{P_A} \cong \left[\mathbf{a}^1, \ldots, \mathbf{a}^N \right] \tag{7.23}$$

$$\mathbf{P_B} \cong \left[\mathbf{b}^1, \ldots, \mathbf{b}^N \right] \tag{7.24}$$

These ensembles can be used to estimate the covariance matrices of the ATs as $\mathbf{C_A} = \mathrm{cov}(\mathbf{P_A})$ and $\mathbf{C_B} = \mathrm{cov}(\mathbf{P_B})$, using the sample covariance operator defined in Equation (2.80), From the covariances, it is possible to compute the Kalman gain factor:

$$\mathbf{K} = \mathbf{C_A} \left[\mathbf{C_A} + \mathbf{C_B} \right]^{-1} \tag{7.25}$$

The samples representing the combined AT can then be computed using the *merging* operation $\otimes$ defined as follows:

$$\mathbf{A} \otimes \mathbf{B} \cong \left[\mathbf{a}^1 + \mathbf{K}(\mathbf{b}^1 - \mathbf{a}^1), \ldots, \mathbf{b}^N + \mathbf{K}(\mathbf{b}^N - \mathbf{a}^N) \right] \tag{7.26}$$

The quaternions of the input ATs must be normalised prior to the merge; the resulting quaternions will be denormalised, in general.

Note that merging is simply a binary operator applied pairwisely to samples with the same index from the input ordered sample sets (aka ensembles). The characteristics of this binary operator are only determined by the global characteristics of both input sample sets, namely, their covariances. The merging operation is just a special case of the EnKF update rule, described in Definition 2.10. As such, it is based on the assumptions that both input sample sets approximate a Gaussian probability distribution, are independent, and the system is linear. Moreover, the covariance of the pairwise differences between an input and the output ordered

sample set estimates the (Gaussian) cross-correlation between the input distribution and the product distribution, which signifies the importance of preserving the order of the sample sets.

Even if the input ATs have a non-Gaussian distribution, the merged AT will tend towards a Gaussian distribution due to the "fuzzy" central limit theorem. Clearly, in such a situation the merge is suboptimal since the product of the input distributions is likely to be non-Gaussian. An alternative approach could, for example, use a density estimator to obtain a parametric approximation of one input AT's probability distribution, and perform importance resampling of the samples from the second AT, as described by (Ong et al., 2006). Although such an approach would, in the limit, provide a theoretically correct estimate of the combined AT, it would, however, lose the order of samples and thus lose the information about correlation between the sample sets, which is crucial for an efficient stochastic map implementation, as will be shown later. Multiple parallel ATs can be merged by successively merging pairs of ATs; the order in which the merges are performed is irrelevant, because the merging operation is both associative and commutative:

$$(\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} \approx \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C}) \tag{7.27}$$

$$\mathbf{A} \otimes \mathbf{B} = \mathbf{B} \otimes \mathbf{A} \tag{7.28}$$

The approximate equality operator ($\approx$) is used to signify the fact that the equation only holds in the limit, as the number of samples grows to infinity.

7.2.2.4 Discussion

Unlike the compounding and inversion operations, merging considers AT samples as simple vectors without any structure. This is a problem because the space of rotations is a manifold, not an Euclidean space. The statistical properties of the merge operation could be further improved by application of the $\boxplus$ operator that maps a local variation in an Euclidean space to a manifold, as described in (Hertzberg et al., 2011). However, this is beyond the scope of this thesis. Although the merging operation as defined in Equation (7.26) is suboptimal, in practice it behaves well even with moderate angular errors.

The representation of ATs by sample sets and the definition of compounding, inversion and merging as operations on samples can be seen as a generalisation of the historical Smith & Cheeseman framework (Smith and Cheeseman, 1986) that operates with ATs represented using means and covariances. Both sample-based compounding and inversion work precisely with arbitrary probability distributions and angular errors in the limit, as the number of samples grows

to infinity. Note that all the three operations can be directly implemented to run asymptotically in $O(N)$ time and space, where N is the number of samples. Furthermore, N does not depend on the covariance of the input distributions, as argued in (Evensen, 2009).

Equivalent AT operations could easily be defined for sample sets with a different parametrisation of coordinate frame rotations, such as Euler angles. The choice of quaternions has, however, several benefits. Firstly, quaternions have no singularities in the poles as Euler angles have, thus avoiding the infamous gimbal lock problem. Secondly, the quaternions are the preferred parametrisation of orientations for statistical purposes (Altmann, 1986), such as the computation of covariance. Thirdly, the AT operations with quaternions do not need to evaluate trigonometric functions, which are computationally expensive.

7.2.3 Ordered Sample Sets Correlation

As mentioned in Section 7.2.2.3, under given assumptions, the result of the merging operation is an ordered sample set, whose correlation to an input ordered sample set can be estimated by pairwisely subtracting the two sample sets and computing the covariance of the difference. Formally, in this context, two ordered sample sets $\mathbf{P_A}$ and $\mathbf{P_B}$ are called *uncorrelated* if:

$$\mathrm{cov}(\mathbf{P_A} - \mathbf{P_B}) \approx \mathrm{cov}(\mathbf{P_A}) + \mathrm{cov}(\mathbf{P_B}) \tag{7.29}$$

where $\mathbf{P_A} - \mathbf{P_B}$ denotes a pairwise difference between two ordered sample sets, which is actually another ordered sample set. The approximate equality ($\approx$) changes to full equality ($=$) in the limit, as the number of samples grows to infinity. The order of the samples plays a key role. For example, consider two correlated ordered samples sets. If one of the sets is randomly permuted, the difference to the other set will suggest the two sets are uncorrelated. Note that the definition above only considers the second moment of the correlation, i.e. the covariance. Assuming the two original distributions, and their correlation, is Gaussian, then the covariance is the theoretically correct estimate of the correlation. However, if these assumptions are not met, the definition might not provide an accurate description of correlation between underlying probability distributions.

The compounding operation does not require the input ATs to be uncorrelated. In fact, compounding of correlated ATs has some desired properties, such as the *eliminability* property for the inversion operation:

$$\mathbf{A} \oplus \mathbf{A^R} = \emptyset \tag{7.30}$$

where Ø denotes an invariant, or zero, AT. However, the merging operation, as defined in Section 7.2.2.3, does require the input ATs to be uncorrelated. Consider three ATs: $\mathbf{X} \in t \rightarrow u$ and $\mathbf{A}, \mathbf{B} \in u \rightarrow v$. With the previously defined merging operation, distributivity of $\oplus$ over $\otimes$ cannot be guaranteed, i.e.

$$\mathbf{X} \oplus (\mathbf{A} \otimes \mathbf{B}) \not\approx (\mathbf{X} \oplus \mathbf{A}) \otimes (\mathbf{X} \oplus \mathbf{B}) \tag{7.31}$$

Given two potentially correlated ATs $\mathbf{C}, \mathbf{D} \in t \rightarrow v$ to merge (represented by two ordered sample sets $\mathbf{P_C}$ and $\mathbf{P_D}$, respectively), suppose the covariances of their errors are composed of a common error component $\mathbf{C_X}$, and independent components $\mathbf{C_A}$ and $\mathbf{C_B}$, respectively:

$$\mathrm{cov}(\mathbf{P_C}) = \mathbf{C_X} + \mathbf{C_A} \tag{7.32}$$

$$\mathrm{cov}(\mathbf{P_D}) = \mathbf{C_X} + \mathbf{C_B} \tag{7.33}$$

$$\mathrm{cov}(\mathbf{P_C} - \mathbf{P_D}) = \mathbf{C_A} + \mathbf{C_B} \tag{7.34}$$

Solving the system of equations leads to:

$$\mathbf{C_X} = \frac{\mathrm{cov}(\mathbf{P_C}) + \mathrm{cov}(\mathbf{P_D}) - \mathrm{cov}(\mathbf{P_C} - \mathbf{P_D})}{2} \tag{7.35}$$

$$\mathbf{C_A} = \mathrm{cov}(\mathbf{P_C}) - \mathbf{C_X} \tag{7.36}$$

$$\mathbf{C_B} = \mathrm{cov}(\mathbf{P_D}) - \mathbf{C_X} \tag{7.37}$$

Plugging $\mathbf{C_A}$ and $\mathbf{C_B}$ into Equations (7.25) and (7.26) leads to a definition of a *revised merging* operation that effectively ignores the correlation between the two input sample sets. This operation is just a special case of the CPI-EnKF update rule provided in Section 5.2.2. In fact, the work presented here represents historically the first derivation of the CPI-EnKF update rule. Note that if the original sample sets $\mathbf{P_C}$ and $\mathbf{P_D}$ are uncorrelated, then $\mathbf{C_X}$ converges to the zero matrix thanks to Equation (7.29) as the number of samples grows to infinity, and the revised merge operation reduces to the standard merge.

The fact that the revised merge is just a generalisation of the standard merge allows to apply the term "merge" and the $\otimes$ symbol to the revised merge in the later text, without any confusion. Note that now the merging operation is idempotent, i.e. $\mathbf{A} \otimes \mathbf{B} \approx \mathbf{A} \otimes \mathbf{B} \otimes \mathbf{B} \otimes ....$ The idempotence guarantees that ATs can be repetitively re-merged, without under or overestimating uncertainty of a combined AT. This feature of the CPI-EnKF update rule is analysed in detail in Section 6.4.2.

7.2.4 Stochastic Map

A mobile robot employs sensors to observe its environment and uses the observed information for navigation. Assuming that spatial observations and navigation decisions can be expressed as operations with ATs, the robot can solve the navigation problem by means of registering observed ATs and querying potentially unobserved ATs. The stochastic map is a data structure that implements two basic procedures: *update* registers a newly observed AT and *query* finds the best estimate of an AT between two given coordinate frames.

The sensor readings are assumed to be expressible by an AT, generated by random sampling from the measurement model. Without loss of generality, assume that the relative positions of coordinate frames in the stochastic map are fixed; if the robot wants to update the location of an existing frame (e.g. its own pose), it can do so by relating a new coordinate frame to the referenced coordinate frame using an AT, and then removing the referenced frame.

In the approach presented in this section, the stochastic map is represented as a forest of star graphs: nodes corresponds to coordinate frames and edges correspond to ATs modelled using ordered Monte Carlo sample sets. Each star in the forest represents a single disjoint component of the map, i.e. a separate sub-map. Although such a distinction is often not necessary, as the map will typically be built sequentially, it is useful to show how two disjoint components can be connected, e.g. in multi-robot SLAM. Figure 7.3 shows an example of such a forest comprised of three star components; the arrows illustrate the direction of the ATs. The root of a star has the role of a global reference frame, similarly to the EKF-SLAM stochastic map (Smith et al., 1990), and the edges store the estimate of ATs to all reachable frames.

If an AT between the root and a leaf frame is queried, the result can be provided directly, possibly utilising a single AT inversion. For example, in Figure 7.3, the $a \to r$ AT is found simply as $\mathbf{A}^{\mathbf{R}}$. The crucial property of the stochastic map is that the ordered sample sets of ATs in a single star component are correlated in such a way that an AT between two arbitrary leaf frames can be found simply by compounding the two ATs connecting those two frames, utilising a single AT inversion. For example, $a \to b$ AT can be computed as $\mathbf{A}^{\mathbf{R}} \oplus \mathbf{B}$. Two different cases arise when adding a new AT to the map: in case I, the newly added AT connects two frames from different components, and in case II, the AT connects two frames in the same component. These two cases are described in detail in the following sections.

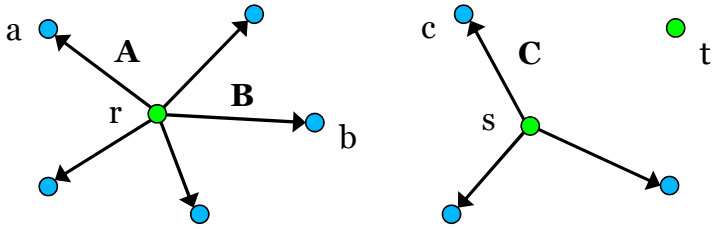


Fig. 7.3: A stochastic map with three disjoint components

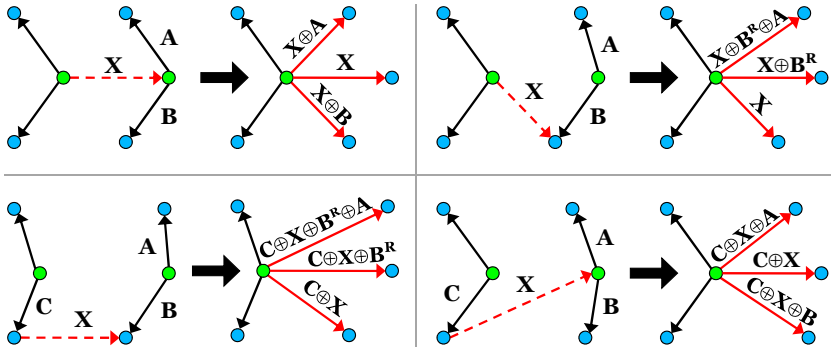


Fig. 7.4: Configurations when joining two components with an AT

7.2.4.1 Case I: Adding an AT connecting two components

If the newly added AT connects two different star components, then it means that the corresponding sub-maps are no longer disjoint and in the stochastic map, the two components will be joined into a single component, using only the compounding and inversion operations. Figure 7.4 shows all four possible situations that can arise when joining two disjoint components with a newly added AT **X**. Note that a component can also be comprised of a single node (see the frame t in Figure 7.5); such an empty component is created whenever a coordinate frame is observed the first time, before the corresponding AT is added to the map.

An important observation is that if both components were *consistent* before the join (i.e. the edges store the best estimates of the ATs, and their sample sets are well-correlated), the joined component will also be consistent thanks to the associativity of compounding from Equation (7.20) and eliminability of the inversion operation from Equation (7.30).

7.2.4.2 Case II: Adding an AT updating a single component

When adding an AT that connects two frames of a single stochastic map component, all ordered sample sets of ATs in that component need to be updated to reflect the new information.

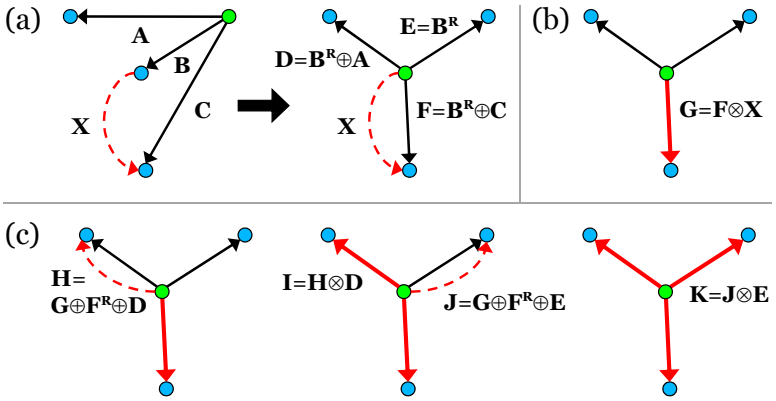


Fig. 7.5: Steps when updating a single component by an AT

Figure 7.5 illustrates the steps of the update process when a new AT **X** is added to a single component of the map. First, the star component might need to be reorganised using the compounding and inversion operations so that the newly added AT goes from its root to a *main* leaf frame (Figure 7.5-a). In the next step, the existing root-to-main AT is merged with the added AT, and the result of the merge replaces the existing AT (see **G** in Figure 7.5-b). Finally, each other root-to-leaf AT in the component is merged with the previously updated root-to-main AT compounded with the old estimate of the main-to-leaf AT (Figure 7.5-c). This way the new information provided by the added AT is propagated to all nodes of the star component, while keeping the structure consistent. Many of the merged AT pairs will typically be correlated, so the revised merging operation must be employed. This also implies that even a newly added AT can be correlated to ATs already present in the stochastic map. In order to generate such an AT from a sensor whose errors are sequentially correlated, it is necessary to generate samples in such a way that the correlation between samples of ATs corresponding to consecutive sensor readings models the sequential correlation characteristics of the sensor error. The details of this procedure are discussed in Chapter 5.

7.2.4.3 Discussion

As mentioned in Section 7.2.2.4, the compounding, inversion and merging operations all run asymptotically in $O(N)$ time and space, where N is the number of samples. The query procedure of the stochastic map processes just one or two edges of a single star component, and only performs a constant number of operations per edge. Also, only a single sample set needs to be kept in memory while processing the query, so the query as a whole executes in $O(N)$ time

and space. On the other hand, the update procedure needs to process each edge a constant number of times, and the number of edges in a star forest asymptotically equals the number of nodes (i.e. coordinate frames), thus enabling the update to run in $O(M \cdot N)$ time where M denotes the number of coordinate frames. For each edge, the stochastic map keeps a single sample set in memory, and both reorganisation of the graph and the update of frames only need a constant number of temporary sample set buffers, so the overall asymptotic space complexity of the stochastic map is also $O(M \cdot N)$.

An important observation is that adding an extra node to a star component does not affect the way in which the queries and updates involving the other nodes are performed. Consequently, the number of samples necessary to represent the ATs is entirely unrelated to the number of nodes in the stochastic map. In other words, the number of samples necessary to accurately represent the map does not grow with the number of spatial features, as in conventional particle filtering methods. The number of samples used to represent ATs is merely a constant parameter that determines the accuracy of the representation of probability distributions and their correlations. Since the N parameter is constant, the asymptotic complexity of a query and update reduces to $O(1)$ and $O(M)$, respectively. This result enables the application of the Monte Carlo stochastic map to much larger environments than the traditional EKF-SLAM with its $O(M^2)$ update complexity. Additionally, to maintain a reasonable performance in practical applications, the robot's perception system might selectively discard coordinate frames corresponding to spatial features that are no longer needed. In contrast to graph-based SLAM methods, the spatial features can be easily removed from the stochastic map, without affecting the estimation of other features. Note that the presented topology is not the only possible way to organise ATs in a stochastic map. The presented organisation was chosen to offer fast queries, while other organisations might favour fast updates by, e.g. building a tree of ATs and delaying updates of ATs that close loops.

7.2.5 Evaluation

This section presents an experimental evaluation of the proposed SLAM approach, with the following goals:

- Evaluate the rate of convergence of representations with a different number of Monte Carlo samples, if all the assumptions are met (i.e. Gaussian distributions and linear errors)
- Evaluate the rate of convergence in a situation where the assumptions are violated, by

adding an angular error to the observations.

- Compare runtime performance of representations with different numbers of Monte Carlo samples.

In the simulated scenario, a mobile robot follows a circular trajectory with a radius of 50 meters, and uses a laser scanner to observe 200 randomly generated point features in its environment. Odometry provides observations of the robot's motion at regular intervals as $(\Delta x, \Delta y, \Delta\theta)$ vectors indicating the displacement of the robot since the previous observation. During every odometry reading, the laser scanner provides an observation of a randomly selected subset of point features available in the front 180-degree field of view, with the maximum range of 30 meters. These observations are provided as two-dimensional vectors (x, y), in the robot's coordinate frame. Both types of observations are subject to white Gaussian noise with standard deviations as follows: $\sigma_x = 0.4m$, $\sigma_y = 0.4m$, $\sigma_{\Delta x} = 0.01m$, $\sigma_{\Delta y} = 0.002m$, and $\sigma_{\Delta\theta}$ varying based on the experiment. Theses values were chosen to emulate errors of a typical laser scanner and odometry, respectively.

7.2.5.1 Convergence

The accuracy of the map is measured as the root mean square (RMS) of the distances between the mean of the position estimates and the ground truth, computed only for observed features. Figure 7.6 shows the accuracy of the map in the situation where odometry has a zero angular error. Before the robot finishes the first revolution of the circular trajectory (approximately at iteration 340), a large loop is closed. Shortly after that all the spatial features have been observed and added to the map, and the accuracy of the maps begins to converge towards zero. The rate of the convergence is faster for maps represented by more Monte Carlo samples. On the other hand, Figure 7.7 demonstrates that if odometry has a non-zero angular error, the map diverges, regardless of the number of samples. This is caused by introduction of non-linear errors by accumulation of relatively small angular errors over time, which then violate the assumptions of the merging operation that all the probability distributions are Gaussian. Similar problems arise with standard EKF-SLAM methods, as they are based on similar assumptions.

7.2.5.2 Runtime Performance

In order to evaluate the runtime performance of the Monte Carlo stochastic map, the map was gradually grown from 0 to 400 frames, while performing 10 AT updates between existing, ran-

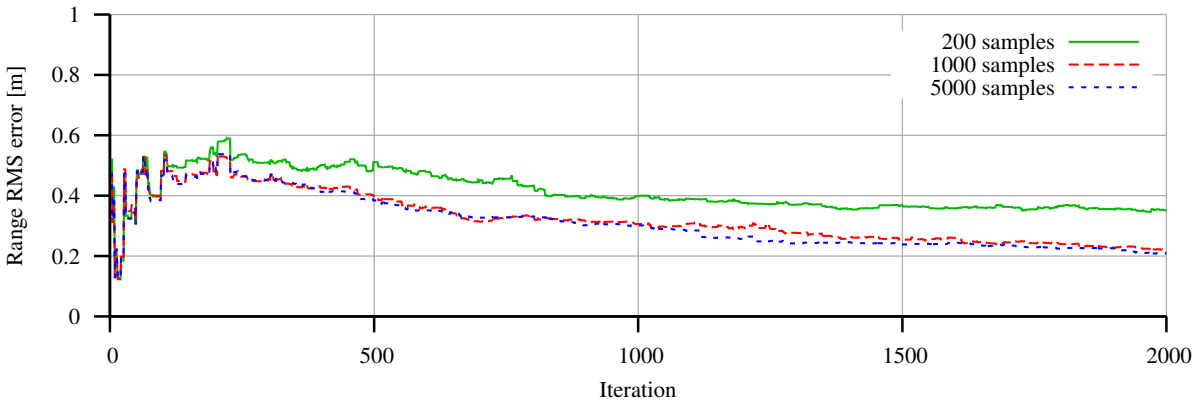


Fig. 7.6: Accuracy of the stochastic map in a scenario with zero angular error in odometry ($\sigma_{\Delta\theta} = 0$).

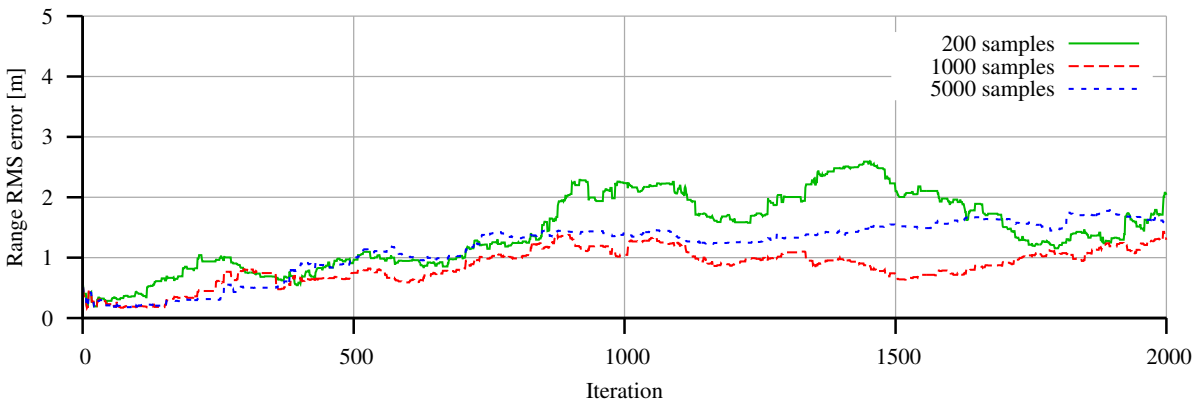


Fig. 7.7: Accuracy of the stochastic map in a scenario with a non-zero angular error in odometry ($\sigma_{\Delta\theta} = 0.1°$).

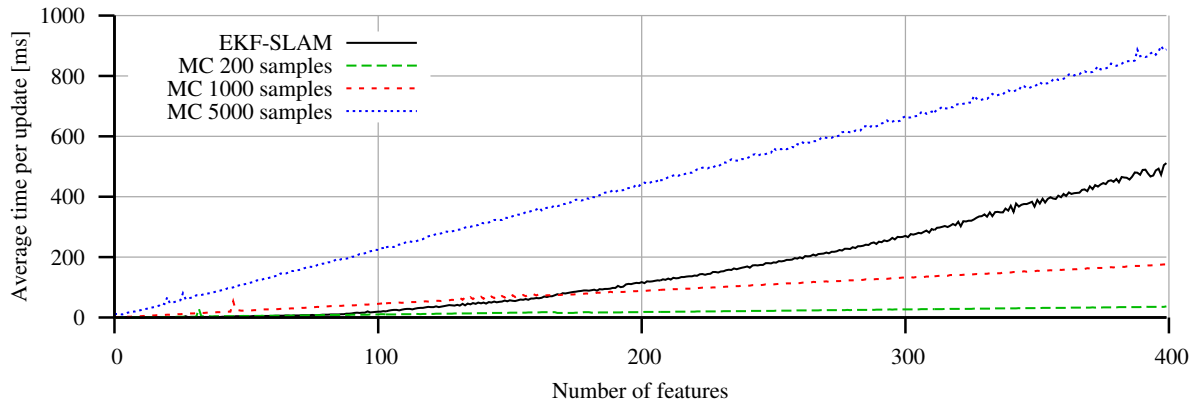


Fig. 7.8: Average runtime performance of the Monte Carlo (MC) stochastic map and the EKF-SLAM stochastic map updates.

domly chosen, frames in every iteration. The test was repeated 10 times, so that the average update time per number of features is computed from 100 samples. Additionally, for demonstration purposes, the same experiment was performed with an implementation of the EKF-SLAM stochastic map, based on (Smith et al., 1990). The results are shown in Figure 7.8. Clearly, the update time of the Monte Carlo stochastic map scales linearly with the number of frames in the map, and with the number of samples used. On the other hand, the graph suggests that EKF-SLAM update times grow quadratically, as expected.

Note that the numbers in Figure 7.8 were obtained with an unoptimised implementation. Since the elementary operations on ATs are isolated simple algebraic computations, there is a significant potential for further optimisation using hardware acceleration, parallel execution, or both.

7.2.6 Conclusion

This section presented a novel algorithm to solve the SLAM problem, based on the CPI-EnKF filter developed in this thesis. The new approach has been evaluated on a simulated data set and its properties compared with the major SLAM paradigms. Its most unique feature is that it scales linearly with the number of spatial features tracked, which enables its application in life-long missions. Additionally, the presented approach disproves the commonly held belief that the number of Monte Carlo samples necessary to accurately represent a posterior probability distribution always grows exponentially with the number of dimensions. Note that the use of the CPI-EnKF rather than the Augmented EnKF in the application presented in this section is due to historical reasons, as this was the first application in which (a special case of) the CPI-EnKF was developed. Nevertheless, the Augmented EnKF would provide equivalent results.

7.3 Global Positioning with a Stand-Alone GPS Receiver

In this section, the CPI-EnKF filter developed in this thesis is applied to a scenario in which a mobile robot estimates its global position by fusing visual odometry data with measurements from a stand-alone GPS receiver, which are affected by a sequentially correlated noise. The main goal of this section is to demonstrate a practical utility of the new EnKF-based correlated estimation methods in real-world problems. In this case, the CPI-EnKF filter enables use of an accurate GPS error model that could not be practically supported using conventional approaches such as state augmentation. The work presented here has been published in (Čurn et al., 2012b).

7.3.1 Introduction

This section demonstrates an application of the CPI-EnKF in a real-world scenario where a mobile robot uses dead reckoning and a low-cost GPS receiver to estimate its global position in real time. The problem of integrating GPS with odometry and other inertial sensors has been studied in depth since the introduction of GPS, and there are two major approaches (Kaplan and Hegarty, 2006): *tightly coupled* and *loosely coupled* integration.

In tightly coupled integration, both the pseudo-range (i.e. distance to satellite) and inertial sensor measurements are processed in a single large Kalman filter. The pseudo-range measurements can be considered as white noise sources.

In low-cost loosely-coupled integration, the GPS receiver is a stand-alone unit that processes pseudo-range measurements using its internal Kalman filter and only reports estimated global coordinates, which, however, cannot be considered as observations affected by a white noise, as their error usually exhibits a strong sequential correlation. If the noise was assumed white, the integration filter might provide overconfident estimates and/or diverge (Julier and Uhlmann, 1997a). The problem lies in the fact that the integration filter cannot model the real underlying system state including variables such as GPS receiver clock bias and clock drift, and it only models a high-level projection of these variables - the global coordinates. Even though the introduction of additional state variables could alleviate the problem, modelling of such non-physical variables is problematic. Another often applied approach is to artificially increase the variance of the observation variables or to ignore certain observations altogether, and thus effectively discard potentially useful information and reduce the convergence rate.

Instead, the approach presented in this section attempts to model the statistical characteristics of noise produced by a stand-alone GPS receiver, and then it uses the new CPI-EnKF to take into account the correlation between the observation error and the state estimate error, which builds up over time. The CPI-EnKF assumes that this correlation is caused by a shared error term between the observation error and the state estimate error, which is to be ignored. This approach is somewhat similar to (Geier et al., 1995), in which the Kalman gain is reduced to take into account the effect of correlations between the state estimate and observation errors. However, their approach uses parameters determined in an ad hoc way for the particular application, and it does not provide any general model of the method. Note that the application presented in this section serves primarily to demonstrate the use and benefits of the CPI-EnKF, and is not a complete solution to the problem outlined.

7.3.2 Stand-Alone GPS Receiver Noise Model

In the long term, a sequence of readings from a stand-alone GPS receiver appears to have a multivariate Gaussian distribution with standard deviations in both (independent) horizontal axes computed as (Kaplan and Hegarty, 2006, p. 331):

$$\sigma_{xy} = \text{HDOP} * \sigma_{\text{UERE}} \tag{7.38}$$

where HDOP is a horizontal dilution of precision reported by the receiver at every reading, and σ_{UERE} is the user-equivalent range error (UERE) constant that summarises the contributing GPS error sources for the specific operation principle of the receiver (the application presented here uses a value of 7.0 m which is a typical value for C/A code receivers without differential correction, and 1.5 m with EGNOS correction (Kaplan and Hegarty, 2006)). However, in the short term, a sequence of GPS readings behaves like a random walk that is stationary around the true position, with a range determined by HDOP.

Formally, at a given time step $t \in \mathbb{N}$, the GPS error in a single horizontal axis can be modelled as a real stochastic process $\{W^t_{\sigma^2_{xy}}\}$ of parameters $t \in \mathbb{N}$ and $\sigma^2_{xy} \in \mathbb{R}$, with the following properties:

$$W^t_0 = 0 \tag{7.39}$$

$$W^t_r - W^t_s \sim \mathcal{N}(0, r - s) \tag{7.40}$$

where $x \sim \mathcal{N}(\mu, \sigma^2)$ denotes that a random vector $x \in \mathbb{R}$ has a normal distribution with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 \in \mathbb{R}$. If the function $r \to W^t_r$ is almost surely continuous, then for a fixed t the sequence $\{W^t_r\}$ is known as a Wiener process. For arbitrary and fixed $r \in \mathbb{R}$, the GPS error at the time step $t + 1$ decorrelates from the previous time step t as:

$$W^{t+1}_r = \varphi W^t_r + \varepsilon^t_r \tag{7.41}$$

$$\varepsilon^t_r \sim \mathcal{N}(0, r(1 - \varphi^2)) \tag{7.42}$$

where $\varphi \in \mathbb{R}$ is a *damping factor* determining the rate of the decorrelation. From this perspective, for a fixed r, the sequence $\{W^t_r\}$ behaves like a wide-sense stationary auto-regressive process of order 1, which is also known as a coloured noise process from Definition 3.3. Note that if $\varphi = 0$, the process decorrelates fully at every time step, and so it becomes a white Gaussian noise process.

In the CPI-EnKF, each observation is represented as an ordered set of N samples (aka ensemble). The goal is to develop a scheme that generates samples with statistical properties of

the stochastic process $\{W_r^t\}$. Such samples could be used as the observation noise compensation terms of the filter, as discussed in Section 5.2.

Assume a fixed time step $t \in \mathbb{N}$, and assume the samples of $W_{r_0}^t, \ldots, W_{r_k}^t$, where $r_0 < \ldots < r_k$, with the desired properties have already been generated, and now the samples of W_s^t for some $s \in \mathbb{R}$ are to be generated. Without loss of generality, assume $r_0 = 0$ and thus the samples $W_{r_0}^t$ are all zeros. If $r_k \leq s$, then the samples of W_s^t can be generated simply by adding a random Gaussian noise to the last set of samples $W_{r_k}^t$, i.e.

$$W_s^t \leftarrow W_{r_k}^t + \mathcal{N}(0, s - r_k) \tag{7.43}$$

If $r_i \leq s < r_j$ for some $i = 0, \ldots, k$ and $j = i + 1, \ldots, k$, then samples W_s^t can be generated by interpolating between the previous samples of $W_{r_i}^t$ and the following samples of $W_{r_j}^t$, while adding a random Gaussian noise as:

$$W_s^t \leftarrow W_{r_i}^t + \alpha(W_{r_j}^t - W_{r_i}^t) + \mathcal{N}(0, (s - r_i)(1 - \alpha)) \tag{7.44}$$

$$\alpha = (s - r_i)/(r_j - r_i) \tag{7.45}$$

The samples of $W_{r_0}^{t+1}, \ldots, W_{r_k}^{t+1}$ for the next time step $t + 1$ can be generated from the current samples of $W_{r_0}^t, \ldots, W_{r_k}^t$ sequentially for $i = 1, \ldots, k$ by applying the following formula:

$$W_{r_i}^{t+1} \leftarrow W_{r_{i-1}}^{t+1} + \varphi(W_{r_i}^t - W_{r_{i-1}}^t) + \mathcal{N}(0, (r_i - r_{i-1})(1 - \varphi^2)) \tag{7.46}$$

Starting at time $t = 0$ with samples of W_0^0 which are all zeros, it can be shown that the recursive sampling scheme described above generates samples that at any time retain the statistical properties defined in Equations (7.39)-(7.42); however, a formal proof of this statement is beyond the scope of this thesis. The sampling scheme described above can be viewed from another perspective: for any $l = 1, \ldots, N$, the sequence of samples at index l generated by the sampling scheme simulates over time the behaviour of a stand-alone GPS receiver, given a true position and HDOP.

From an implementation perspective, at any time step t the system maintains a linked-list of sample sets generated for all unique values σ_{xy} observed thus far (i.e. the number of unique HDOP values reported by the GPS receiver). Whenever samples for a new GPS observation error are required, first the standard deviation σ_{xy} of the GPS error is computed using Equation (7.38) and the HDOP value reported by the GPS receiver, and then the samples of $W_{\sigma_{xy}^2}^t$ are generated using the sampling scheme described above. These samples can be readily used in the CPI-EnKF update rule as the observation noise compensation term, as described in Section 5.2. In filtering

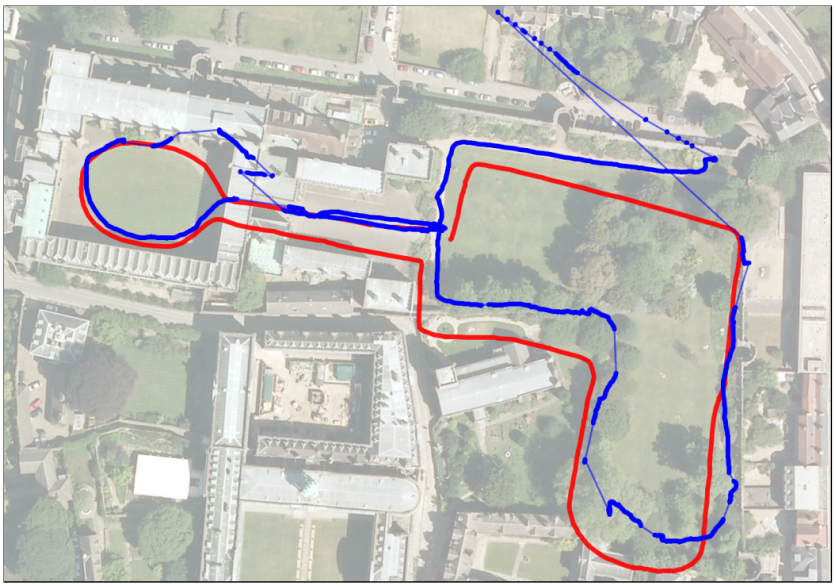


Fig. 7.9: The New College Dataset (Smith et al., 2009) - visual odometry is shown in red and GPS data in blue.

scenarios with unbounded time the requirement to keep sample sets for all unique HDOP values can cause significant memory and performance overheads, thus sample sets corresponding to the HDOP values not observed for a long time need to be periodically discarded. In the scenario discussed below, this issue was not a concern.

Note that in order to implement a similar estimation system with the conventional Kalman filter, the system state would have to be augmented with terms corresponding to all the unique HDOP values reported by the received in the past. Considering the quadratic complexity of the update step of the Kalman filter, such a system would become computationally very expensive and most likely inapplicable in practice. On the other hand, with the CPI-EnKF the computationally cost can be kept relatively low.

7.3.3 Scenario

The evaluation of the approach uses data from the New College Dataset (Smith et al., 2009), where a Segway-based mobile robot equipped with a number of sensors traverses an outdoor environment of the New College campus at the University of Oxford. The evaluation only considers the data provided by the CSI Seres 5Hz GPS receiver with a DGPS correction using WAAS/EGNOS, and visual odometry (VO) data computed from the LadyBug 2 panoramic camera with a 20 Hz frame rate (see Figure 7.9). The raw odometry integrated from the wheel

counters was not used due to its high inaccuracy. For clarity, only the last circumnavigation of the campus is shown in all the figures. The satellite imagery was taken from Google Maps.

The dynamical system is modelled using 3 state variables: latitude and longitude (both expressed as a distance in meters from a fixed coordinate origin nearby), and the yaw angle. The ensemble (size $N = 1000$) is propagated in time using the VO readings with an added random Gaussian noise (standard deviation ca. 0.55 cm in both the latitude and longitude, and ca. 0.1° in the yaw angle). All the valid GPS readings are used to update the ensemble. The disparity in time between the GPS and VO readings (of maximum 50 ms) is neglected.

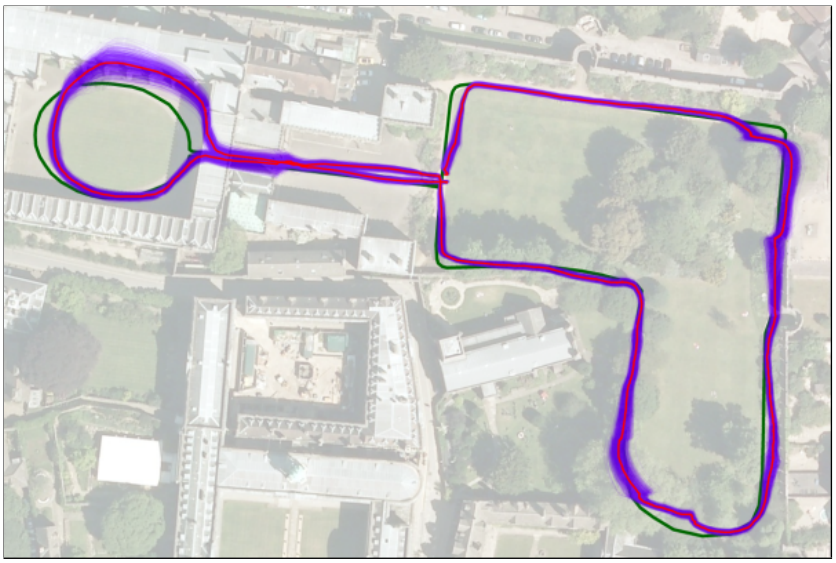
In this evaluation, the performance of two estimators is compared: the conventional EnKF with GPS errors sampled as a white Gaussian noise with variance given by the HDOP using Equation (7.38), and the CPI-EnKF with GPS errors sampled using the scheme described in Section 7.3.2, with a damping factor that decorrelates the GPS readings after 30 seconds (up to 1%, 5 Hz rate, $\varphi = \sqrt[5\times 30]{0.01} \approx 0.970$). The performance is compared against ground truth that was defined by manually drawing a polyline that follows the true robot's trajectory, based on the satellite imagery of the campus and images from the panoramic camera.

7.3.4 Results

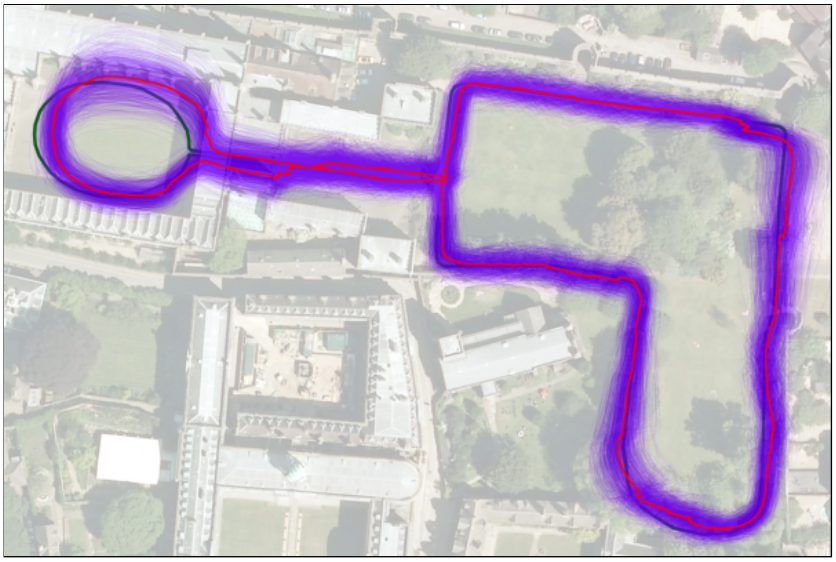
The estimates of the robot's trajectory from both the conventional EnKF and the CPI-EnKF are shown in Figure 7.10. In areas with good GPS coverage, the estimated trajectory from both filters closely follows the ground truth, which also indicates that the satellite images are aligned correctly to the global coordinates. In areas with poor GPS coverage, both trajectories often deviate from the ground truth.

In the case of the conventional EnKF, often when the estimated trajectory deviates significantly from the ground truth, it does so overconfidently. In these areas, basically none of the ensemble members (i.e. samples of the estimated robot's pose) is anywhere near the ground truth, indicating that the probability of the robot being in the ground truth position is practically zero. Clearly, these estimates do not reflect the reality. If such estimates were to affect some safety-critical decisions of the robot, the consequences could be severe.

In the case of the CPI-EnKF, the estimated trajectory is similar to that of the traditional EnKF, however, the uncertainty associated with the estimates is far more realistic. The ensemble members cover the ground truth, giving a non-zero probability to the eventuality that the robot is actually in the ground truth position. If the models of the system and the noises are correct,



(a) EnKF



(b) CPI-EnKF

Fig. 7.10: Estimated trajectories of the robot from the New College Dataset. The ground truth is shown in green, the mean position estimates in red, and the uncertainty associated with the estimates in violet.

the provided estimates are guaranteed to be consistent, in the limit of an infinite ensemble.

7.3.5 Conclusion

The CPI-EnKF filter proposed in this thesis has been evaluated in a real-world scenario where a global position of a mobile robot is estimated from visual odometry and a sequence of GPS read-

ings affected by a sequentially correlated noise process, using a sophisticated sampling scheme that emulates such a noise process. The evaluation shows that the CPI-EnKF provides much more realistic probability estimates than the conventional EnKF. As it has been discussed, implementation of such an estimation system using a conventional Kalman filter would make the estimation computationally very expensive, and thus inapplicable in practice.

It needs to be noted that the application of the CPI-EnKF presented in this section violates the formal assumption that the observation model is an invertible matrix, as discussed in Section 5.2.2. Therefore, a more correct solution would be to apply the Augmented CPI-EnKF instead of the CPI-EnKF. However, in order to keep this section consistent with (Čurn et al., 2012b), and also because the estimates provided by the CPI-EnKF are accurate enough, the Augmented EnKF was not used in this section.

7.4 Chapter Summary

This chapter presented three practical applications of the EnKF-based algorithms for correlated estimation problems developed in this thesis, in particular the CPI-EnKF and the Augmented EnKF filters, to three distinct problems in the larger field of robotic localisation. In each of the problems discussed, the new filters facilitate novel solutions that provide some new, qualitatively better, features compared to the existing state-of-the-art methods. In general, these features stem from the favourable linear computational complexity associated with the Monte Carlo representation of probability distributions, and from the novel ability to fuse such distributions even if they are correlated.

Chapter 8

Conclusions and Future Work

This chapter summarises the achievements and contributions of this thesis, and discusses possible paths for future work in the area.

8.1 Achievements

This thesis had an ambitious goal of exploring the applicability of the Monte Carlo method to linear estimation problems affected by correlations in the errors. The Monte Carlo method has already proven its merit in linear state estimation, enhancing the most prominent linear estimation algorithm, the Kalman filter, with several important features such as a linear space and time complexity and a support of non-linear process and observation models with arbitrary accuracy. The resulting algorithm, the Ensemble Kalman filter (EnKF), has proven to be an invaluable tool in the Earth sciences, for example, it became a de facto standard tool for data assimilation in numerical weather forecasting.

The state of the art review presented in this thesis identified two primary origins of a correlation in linear estimation problems - correlated noises and common past information, and provided the most comprehensive overview of applicable Kalman filter-based estimation approaches to date. The most universal of these approaches, the Covariance Intersection (CI) algorithm, provides an alternative update rule, which can account for an unknown correlation between the errors associated with the state estimate and the observation. The universality of the CI update rule enables its application to a wide range of problems, in particular to a distributed data fusion in sensor networks.

The thesis then developed its main contribution, a generalisation of the EnKF filter, which

enables it to operate in the presence of correlations in the errors, regardless of whether they are caused by correlated noises or common past information. In particular, the thesis presents a derivation of two new universal update rules, referred to as the Common Past-Invariant Ensemble Kalman filter (CPI-EnKF) and the Augmented EnKF, which both provide consistent state estimates even in the presence of a correlation between the errors associated with the state estimate and the observation. This is only possible due to the unique nature of the Monte Carlo representation of probability distributions inherent to the EnKF, which enables estimation of the appropriate cross-covariance from random samples.

Numerical properties of all the new methods have been thoroughly evaluated in canonical instances of all the types of correlated estimation problems discussed in the state of the art review. The results indicate that the CPI-EnKF and the Augmented EnKF provide consistent estimates in all of these types of problems. Moreover, in the case of estimation problems affected by common past information, which is characteristic of sensor networks, the new update rules provide more accurate state estimates than the only comparable algorithm - the CI. As such, both the CPI-EnKF and the Augmented EnKF represent valuable tools for distributed data fusion.

In order to demonstrate the practical utility of the new methods developed in this thesis to real-world problems, the CPI-EnKF and the Augmented EnKF filters have been applied to three distinct problems in the larger field of robotic localisation: cooperative localisation, simultaneous localisation and mapping (SLAM) and global satellite-based positioning. The filters enable development of new approaches that provide unique features compared to state-of-the-art approaches in the respective areas. Moreover, this work also practically demonstrates that the EnKF has the merit in real-time robotics applications, not only in the Earth sciences.

Considering that the Augmented EnKF is more general and numerically stable than the CPI-EnKF, the main contribution of this thesis can be summarised by Equation (5.62):

$$\mathbf{K} \;=\; \mathrm{cov}(\mathbf{X}, \mathbf{HX} - \mathbf{Z})\; \mathrm{cov}(\mathbf{HX} - \mathbf{Z})^{-1}$$

which represents the only part of the update rule that is different from the original EnKF.

8.2 Future Work

One of the principal limitations of the EnKF as well as its variants developed in this thesis is the theoretical assumption that all the probability distributions involved are Gaussian. Although the

filters provide reasonable state estimates even if this assumption is not exactly satisfied, typically even with a higher accuracy than both the EKF and UKF, the filters are not suitable to problems characterised by a presence of multimodal distributions (i.e. probability distributions with two or more distinct peaks). A possible solution would be to extend the filters to represent state estimates using more than one ensemble, effectively creating an EnKF variant of the multivariate Gaussian mixture model.

Another problem of the methods developed in this thesis, as well as all other variants of the Kalman filter, is the irreversibility of assimilation of sensor measurements. Once a sensor measurement is used to update the state estimate, it cannot be easily removed, shall the measurement be found incorrect later. For example, such a feature is useful for implementing a robust data association in robotics. A naive approach to address this problem would be to maintain multiple state estimates at the same time, where each estimate is updated by a different set of (alternative) observations. Unfortunately, this approach leads to an exponential explosion of the number of state estimates maintained. With the EnKF, however, it might be possible to develop a scheme that would reduce the ensemble sizes with each such division, and thus manage the complexity of the problem, trading it off for the filter's accuracy.

Both the new update rules developed in this thesis, the CPI-EnKF and the Augmented EnKF, are based on different premises and are not algebraically equivalent, except in one-dimensional systems. However, in all the scenarios evaluated, the filters provide practically identical state estimates. It remains an open question in which practical problems, if any, will these filters provide substantially different state estimates.

The thesis presented three different practical applications of the CPI-EnKF and the Augmented EnKF filters, with a focus on the area of robotic localisation. However, the filters can be applied and potentially provide interesting benefits in many other areas, for example, in financial analysis. It remains to be seen where the filters will be applied and what benefits will they offer.

Bibliography

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications magazine*, 40(8):102–114.

Alighanbari, M. and How, J. P. (2006). An unbiased Kalman consensus algorithm. In *Proceedings of the 2006 American Control Conference (ACC 2006)*.

Alriksson, P. and Rantzer, A. (2006). Distributed Kalman filtering using weighted averaging. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2006)*.

Altmann, S. L. (1986). *Rotations, Quaternions, and the Double Group*. Oxford University Press.

Anderson, B. D. and Moore, J. B. (1979). *Optimal Filtering*. Prentice-Hall, Inc.

Bailey, T., Julier, S., and Agamennoni, G. (2012). On conservative fusion of information with unknown non-Gaussian dependence. In *Proceedings of the 15th International Conference on Information Fusion (FUSION 2012)*.

Barooah, P., Russell, W. J., and Hespanha, J. P. (2010). Approximate distributed Kalman filtering for cooperative multi-agent localization. In *Distributed Computing in Sensor Systems*, pages 102–115. Springer.

Brehard, T. and Krishnamurthy, V. (2007). Optimal data incest removal in Bayesian decentralized estimation over a sensor network. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*.

Bromiley, P. A. (2003). Products and convolutions of Gaussian distributions. Technical report, University of Manchester. Tina Memo No. 2003-003.

Brown, R. G. and Hwang, P. Y. C. (2012). *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises, 4th Edition*. Wiley.

Bryson, A. E. and Henrikson, L. J. (1967). Estimation using sampled-data containing sequentially correlated noise. Technical report, Harvard University, Cambridge, Massachusetts.

Bryson, A. E. and Johansen, D. E. (1965). Linear filtering for time-varying systems using measurements containing colored noise. *IEEE Transactions on Automatic Control*, 10(1):4–10.

Bucy, R. and Joseph, P. (1968). *Filtering for stochastic processes with applications to guidance*. Interscience tracts in pure and applied mathematics. Interscience Publishers.

Carli, R., Chiuso, A., Schenato, L., and Zampieri, S. (2008). Distributed Kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 26(4):622–633.

Cetin, M., Chen, L., Fisher III, J. W., Ihler, A. T., Moses, R. L., Wainwright, M. J., and Willsky, A. S. (2006). Distributed fusion in sensor networks. *IEEE Signal Processing Magazine*, 23(4):42–55.

Chang, D.-C. and Wu, W.-R. (2001). Maneuvering target tracking with high-order correlated noise - a multirate Kalman filtering approach. *Wireless Personal Communications*, 17(1):103–123.

Chong, C.-Y. and Mori, S. (2004). Graphical models for nonlinear distributed estimation. In *Proceedings of the 7th International Conference on Information Fusion (FUSION 2004)*.

Crassidis, J. L. and Junkins, J. L. (2011). *Optimal Estimation of Dynamic Systems, Second Edition*. CRC Press.

De Gunst, M. C. M., Künsch, H. R., and Schouten, J. G. (2001). Statistical analysis of ion channel data using hidden Markov models with correlated state-dependent noise and filtering. *Journal of the American Statistical Association*, 96(455):805–815.

Dellaert, F. and Kaess, M. (2006). Square root sam: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Reasearch*, 25:2006.

Dodin, P. and Nimier, V. (2002). Distributed tracking systems and their optimal inference topology. In *Proceedings of the 5th International Conference on Information Fusion (FUSION 2002)*.

Doucet, A. and Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. In *Handbook of Nonlinear Filtering*. Oxford University Press.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping (SLAM): Part I the essential algorithms. *IEEE Robotics & Automation Magazine*, 13(2):99–110.

Durrant-Whyte, H., Stevens, M., and Nettleton, E. (2001). Data fusion in decentralised sensing networks. In *Proceedings of the 4th International Conference on Information Fusion (FUSION 2001)*.

Durrant-Whyte, H. F., Rao, B., and Hu, H. (1990). Toward a fully decentralized architecture for multi-sensor data fusion. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation (ICRA 1990)*.

Eckhardt, R. (1987). Stan Ulam, John von Neumann, and the Monte Carlo method. *Los Alamos Science*, 15:131–136.

Eliazar, A. and Parr, R. (2003). DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5):10143–10162.

Evensen, G. (2003). The Ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean dynamics*, 53(4):343–367.

Evensen, G. (2009). *Data Assimilation: The Ensemble Kalman Filter, 2nd Edition*. Springer.

Fox, D., Burgard, W., Kruppa, H., and Thrun, S. (2000). A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344.

Fränken, D. and Hüpper, A. (2005). Improved fast covariance intersection for distributed data fusion. In *Proceedings of the 8th International Conference on Information Fusion (FUSION 2005)*.

Fraser, D. C. (1967). *A new technique for the optimal smoothing of data*. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics.

Frese, U. (2006). Treemap: An O(log n) algorithm for indoor simultaneous localization and mapping. *Autonomous Robots*, 20:25–42.

Gazit, R. (1997). Digital tracking filters with high order correlated measurement noise. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1):171–177.

Geier, G. J., Heshmati, A., Johnson, K. G., and McLain, P. W. (1995). Position and velocity estimation system for adaptive weighting of GPS and dead-reckoning information. US Patent 5416712.

Gelb, A. (1974). *Applied Optimal Estimation*. MIT Press.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET.

Grewal, M. S. and Andrews, A. P. (2011). *Kalman Filtering: Theory and Practice Using MATLAB, Third Edition*. Wiley.

Grime, S. and Durrant-Whyte, H. (1994). Data fusion in decentralized sensor networks. *Control Engineering Practice*, 2(5):849–863.

Grime, S., Durrant-Whyte, H., and Ho, P. (1992). Communication in decentralized data-fusion systems. In *Proceedings of the 1992 American Control Conference (ACC 1992)*.

Grisetti, G., Stachniss, C., and Burgard, W. (2005). Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*.

Guivant, J. and Nebot, E. (2001). Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, 17:242–257.

Guu, J.-A. and Wei, C.-H. (1991). Tracking technique for manoeuvring target with correlated measurement noises and unknown parameters. *IEE Proceedings - Part F: Radar & Signal Processing*, 138(3):278–288.

Hammersley, J. M. and Morton, K. W. (1954). Poor man's Monte Carlo. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 23–38.

Hertzberg, C., Wagner, R., Frese, U., and Schröder, L. (2011). Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77.

Howard, A., Mataric, M., and Sukhatme, G. (2003). Putting the 'I' in 'team': an ego-centric approach to cooperative localization. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA 2003)*.

Howard, A., Matark, M., and Sukhatme, G. (2002). Localization for mobile robot teams using maximum likelihood estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*.

Huang, G., Trawny, N., Mourikis, A., and Roumeliotis, S. (2009). On the consistency of multi-robot cooperative localization. In *Proceedings of Robotics: Science and Systems (RSS 2009)*.

Huang, S., Wang, Z., and Dissanayake, G. (2008). Sparse local submap joining filter for building large-scale maps. *IEEE Transactions on Robotics*, 24(5):1121–1130.

Jazwinski, A. (2007). *Stochastic Processes and Filtering Theory*. Dover Publications.

Jiang, P., Zhou, J., and Zhu, Y. (2010). Globally optimal Kalman filtering with finite-time correlated noises. In *Proceedings of the 49th IEEE Conference on Decision and Control (CDC 2010)*.

Julier, S. and Uhlmann, J. (1996). A general method for approximating nonlinear transformations of probability distributions. Technical report, University of Oxford.

Julier, S. and Uhlmann, J. (1997a). A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the American Control Conference (ACC 1997)*, volume 4.

Julier, S. and Uhlmann, J. (2001a). General decentralized data fusion with covariance intersection (CI). In *Handbook of Multisensor Data Fusion*, chapter 12. CRC Press.

Julier, S. and Uhlmann, J. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422.

Julier, S. J. and Uhlmann, J. K. (1997b). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of the 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls (AeroSense 1997)*.

Julier, S. J. and Uhlmann, J. K. (2001b). Simultaneous localisation and map building using split covariance intersection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*.

Kaess, M., Ranganathan, A., and Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics (TRO)*, 24(6):1365–1378.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45.

Kalman, R. E. (1963). New methods in Wiener filtering theory. In *Proceedings of the First Symposium on Engineering Applications of Random Function Theory and Probability*, pages 270–388. John Wiley and Sons.

Kaplan, E. and Hegarty, C. (2006). *Understanding GPS: Principles and Applications, 2nd Edition*. Artech House Mobile Communications Series. Artech House.

Karam, N., Chausse, F., Aufrere, R., and Chapuis, R. (2006). Localization of a group of communicating vehicles by state exchange. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*.

Kitagawa, G. (1987). Non-Gaussian statespace modeling of nonstationary time series. *Journal of the American Statistical Association*, 82(400):1032–1041.

Kumar, A. and Crassidis, J. L. (2007). Colored-noise Kalman filter for vibration mitigation of position/attitude estimation systems. In *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit (GNC 2007)*.

Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). G2o: A general framework for graph optimization. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA 2011)*.

Lee, E., Yang, S., Oh, S., and Gerla, M. (2009). RF-GPS: RFID assisted localization in VANETs. In *Proceedings of the 6th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS 2009)*.

Leonard, J. J. and Jacob, H. (1999). A computationally efficient method for large-scale concurrent mapping and localization. In *Proceedings of the Ninth International Symposium on Robotics Research (ISRR 1999)*.

Leung, K., Barfoot, T., and Liu, H. (2009). Decentralized localization for dynamic and sparse robot networks. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*.

Li, F., Zhou, J., and Wu, D. (2011). Optimal filtering for systems with finite-step autocorrelated noises and multiple packet dropouts. *Aerospace Science and Technology*.

Li, H. and Nashashibi, F. (2012). Cooperative multi-vehicle localization using split covariance intersection filter. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV 2012)*.

Li, X. R., Han, C., and Wang, J. (2000). Discrete-time linear filtering in arbitrary noise. In *Proceedings of the 39th IEEE Conference on Decision and Control (CDC 2000)*.

Lu, F. and Milios, E. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349.

Lynch, N. A. (1996). *Distributed Algorithms*. Morgan Kaufmann Publishers.

Mandel, J. (2006). Efficient implementation of the Ensemble Kalman filter. Technical report, University of Colorado at Denver and Health Sciences Center.

Marinescu, D., Čurn, J., Bouroche, M., and Cahill, V. (2012). On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach. In *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems (ITSC 2012)*.

Marinescu, D., Čurn, J., Slot, M., Bouroche, M., and Cahill, V. (2010). An active approach to guaranteed arrival times based on traffic shaping. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010)*.

Martinelli, A. (2007). Improving the precision on multi robot localization by using a series of filters hierarchically distributed. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*.

Maybeck, P. S. (1979). *Stochastic Models, Estimation and Control*, volume 1. Academic Press.

McElhoe, B. A. (1966). An assessment of the navigation and course corrections for a manned flyby of Mars or Venus. *IEEE Transactions on Aerospace and Electronic Systems*, AES-2(4):613–623.

McLaughlin, S., Krishnamurthy, V., and Evans, R. J. (2004). Bayesian network model for data incest in a distributed sensor network. In *Proceedings of the 7th International Conference on Information Fusion (FUSION 2004)*.

McLaughlin, S. P., Evans, R. J., and Krishnamurthy, V. (2005). A graph theoretic approach to data incest management in network centric warfare. In *Proceedings of the 8th International Conference on Information Fusion (FUSION 2005)*.

Meinhold, R. J. and Singpurwalla, N. D. (1983). Understanding the Kalman filter. *The American Statistician*, 37(2):123–127.

Minkler, G. and Minkler, J. (1993). *Theory and Application of Kalman Filtering*. Magellan Book Company.

Montemerlo, M. and Thrun, S. (2007). *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics (Springer Tracts in Advanced Robotics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Mourikis, A. and Roumeliotis, S. (2006). Performance analysis of multirobot cooperative localization. *IEEE Transactions on Robotics*, 22(4):666–681.

Nerurkar, E., Roumeliotis, S., and Martinelli, A. (2009). Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*.

Niehsen, W. (2002). Information fusion based on fast covariance intersection filtering. In *Proceedings of the 5th International Conference on Information Fusion (FUSION 2002)*.

Nieto, F. H. and Guerrero, V. M. (1995). Kalman filter for singular and conditional state-space models when the system state and the observational error are correlated. *Statistics & Probability Letters*, 22(4):303 – 310.

O'Hara, N., Slot, M., Marinescu, D., Čurn, J., Yang, D., Asplund, M., Bouroche, M., Clarke, S., and Cahill, V. (2012). MDDSVsim: An integrated traffic simulation platform for autonomous vehicle research. In *The International Workshop on Vehicular Traffic Management for Smart Cities (VTM 2012)*.

Olfati-Saber, R. (2009). Kalman-Consensus filter: Optimality, stability, and performance. In *Proceedings of the 48th IEEE Conference on Decision and Control, held jointly with the 28th Chinese Control Conference (CDC/CCC 2009)*.

Olfati-Saber, R., Fax, J. A., and Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233.

Olson, E., Leonard, J., and Teller, S. (2006). Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA 2006)*.

Ong, L.-L., Upcroft, B., Ridley, M., Bailey, T., Sukkarieh, S., and Durrant-Whyte, H. (2006). Consistent methods for decentralised data fusion using particle filters. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*.

Paz, L. M., Jensfelt, P., Tardós, J. D., and Neira, J. (2007). EKF SLAM updates in O(n) with divide and conquer SLAM. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA 2007)*.

Petovello, M., O'Keefe, K., Lachapelle, G., and Cannon, M. (2005). Quantifying ambiguity resolution performance in the presence of time-correlated measurement errors using geometricbased techniques. In *Proceedings of the 61st Institute of Navigation Annual Meeting*.

Petovello, M., O'Keefe, K., Lachapelle, G., and Cannon, M. (2009). Consideration of time-correlated errors in a Kalman filter applicable to GNSS. *Journal of Geodesy*, 83:51–56.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (2007). *Numerical Recipes: The Art of Scientific Computing. Third Edition*. Cambridge University Press.

Ranganathan, A., Kaess, M., and Dellaert, F. (2007). Loopy SAM. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI 2007)*.

Rao, B., Durrant-Whyte, H., and Sheen, J. (1993). A fully decentralized multi-sensor system for tracking and surveillance. *The International Journal of Robotics Research*, 12(1):20–44.

Ren, W., Beard, R. W., and Kingston, D. B. (2005). Multi-agent Kalman consensus with relative uncertainty. In *Proceedings of the 2005 American Control Conference (ACC 2005)*.

Ren, W., Beard, R. W., and McLain, T. W. (2004). Coordination variables and consensus building in multiple vehicle systems. In *Cooperative Control*, volume 309 of *Lecture Notes in Control and Information Science*, pages 171–188. Springer.

Rogers, S. R. (1987). Alpha-beta filter with correlated measurement noise. *IEEE Transactions on Aerospace and Electronic Systems*, 23(4):592–594.

Roumeliotis, S. and Bekey, G. (2002). Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795.

Roumeliotis, S. and Rekleitis, I. (2004). Propagation of uncertainty in cooperative multirobot localization: Analysis and experimental results. *Autonomous Robots*, 17(1):41–54.

Simon, D. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley.

Skolnik, M. (2008). *Radar Handbook, Third Edition*. Electronics electrical engineering. McGraw-Hill.

Smith, G., Schmidt, S., and McGee, L. (1962). Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. Technical report, National Aeronautics and Space Administration. NASA TR R-135.

Smith, M., Baldwin, I., Churchill, W., Paul, R., and Newman, P. (2009). The New College vision and laser data set. *The International Journal of Robotics Research*, 28(5):595–599.

Smith, R., Self, M., and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, 4:167–193.

Smith, R. C. and Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, 5(4).

Sorenson, H. W. (1985). *Kalman Filtering: Theory and Application*. IEEE Press selected reprint series. IEEE Press.

Spanos, D. P. and Murray, R. M. (2005). Distributed sensor fusion using dynamic consensus. In *Proceedings of the 16th IFAC World Congress (IFAC 2005)*.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

Thrun, S. and Urmson, C. (2011). Self-driving cars. Keynote Address at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011).

Uhlmann, J. K. (1995). *Dynamic map building and localization: New theoretical foundations*. PhD thesis, University of Oxford.

Van Der Merwe, R., Doucet, A., De Freitas, N., and Wan, E. (2000). The unscented particle filter. Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department.

Van Der Merwe, R. and Wan, E. (2001). The square-root unscented Kalman filter for state and parameter-estimation. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*.

Čurn, J., Marinescu, D., and Cahill, V. (2010). A flexible approach to management and processing of collaborative vehicular perception data. In *Proceedings of the Workshop on Emergent Cooperative Technologies in Intelligent Transportation Systems (ECTITS 2010) at the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010)*.

Čurn, J., Marinescu, D., and Cahill, V. (2012a). On the Monte Carlo representation of uncertain spatial constraints. *Procedia Engineering*, 41(0):37–46. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).

Čurn, J., Marinescu, D., Lacey, G., and Cahill, V. (2012b). Estimation with non-white Gaussian observation noise using a generalised Ensemble Kalman filter. In *Proceedings of the IEEE International Symposium on Robotic and Sensors Environments (ROSE 2012)*.

Čurn, J., Marinescu, D., O'Hara, N., and Cahill, V. (2013). Data incest in cooperative localisation with the Common Past-Invariant Ensemble Kalman filter. In *Proceedings of the 16th International Conference on Information Fusion (FUSION 2013)*. To appear.

Wan, E. A. and Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (AS-SPCC 2000)*.

Wang, K., Li, Y., and Rizos, C. (2012). Practical approaches to Kalman filtering with time-correlated measurement errors. *IEEE Transactions on Aerospace and Electronic Systems*, 48(2):1669–1681.

Wimmel, H. (1992). *Quantum Physics & Observed Reality: A Critical Interpretation of Quantum Mechanics*. World Scientific.

Wu, W.-R. and Chang, D.-C. (1996). Maneuvering target tracking with colored noise. *IEEE Transactions on Aerospace and Electronic Systems*, 32(4):1311–1320.

Xiao, L., Boyd, S., and Lall, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*.