Linköping studies in science and technology. Licentiate Thesis.
No. 1613

Kalman Filters for Nonlinear Systems and Heavy-Tailed Noise

Michael Roth

REGLERTEKNIK
AUTOMATIC CONTROL
LINKÖPING

Division of Automatic Control
Department of Electrical Engineering
Linköping University, SE-581 83 Linköping, Sweden
`http://www.control.isy.liu.se`
`roth@isy.liu.se`

Linköping 2013

This is a Swedish Licentiate's Thesis.

Swedish postgraduate education leads to a Doctor's degree and/or a Licentiate's degree.
A Doctor's Degree comprises 240 ECTS credits (4 years of full-time studies).
A Licentiate's degree comprises 120 ECTS credits,
of which at least 60 ECTS credits constitute a Licentiate's thesis.

Linköping studies in science and technology. Licentiate Thesis.
No. 1613

Kalman Filters for Nonlinear Systems and Heavy-Tailed Noise

Michael Roth

roth@isy.liu.se
www.control.isy.liu.se
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping
Sweden

ISBN 978-91-7519-535-3 ISSN 0280-7971 LIU-TEK-LIC-2013:47

Printed by LiU-Tryck, Linköping, Sweden 2013

Dedicated to all vegetarians.

Abstract

This thesis is on filtering in state space models. First, we examine approximate Kalman filters for nonlinear systems, where the optimal Bayesian filtering recursions cannot be solved exactly. These algorithms rely on the computation of certain expected values. Second, the problem of filtering in linear systems that are subject to heavy-tailed process and measurement noise is addressed.

Expected values of nonlinearly transformed random vectors are an essential ingredient in any Kalman filter for nonlinear systems, because of the required joint mean vector and joint covariance of the predicted state and measurement. The problem of computing expected values, however, goes beyond the filtering context. Insights into the underlying integrals and useful simplification schemes are given for elliptically contoured distributions, which include the Gaussian and Student's *t* distribution. Furthermore, a number of computation schemes are discussed. The focus is on methods that allow for simple implementation and that have an assessable computational cost. Covered are basic Monte Carlo integration, deterministic integration rules and the unscented transformation, and schemes that rely on approximation of involved nonlinearities via Taylor polynomials or interpolation. All methods come with realistic accuracy statements, and are compared on two instructive examples.

Heavy-tailed process and measurement noise in state space models can be accounted for by utilizing Student's *t* distribution. Based on the expressions for conditioning and marginalization of *t* random variables, a compact filtering algorithm for linear systems is derived. The algorithm exhibits some similarities with the Kalman filter, but involves nonlinear processing of the measurements in form of a squared residual in one update equation. The derived filter is compared to state-of-the-art filtering algorithms on a challenging target tracking example, and outperforms all but one optimal filter that knows the exact instances at which outliers occur.

The presented material is embedded into a coherent thesis, with a concise introduction to the Bayesian filtering and state estimation problems; an extensive survey of available filtering algorithms that includes the Kalman filter, Kalman filters for nonlinear systems, and the particle filter; and an appendix that provides the required probability theory basis.

Populärvetenskaplig sammanfattning

När Kalman år 1960 presenterade sin lösning för filtrering och prediktion i dynamiska system, så startade han en helt ny era inom signalbehandling. Kalmanfiltret kom helt rätt i tiden och fick en ovanligt snabb spridning. Dels gick det att implementera på den tidens datorer, dels så löste det akuta problem i Apolloprojektet, vilket blev dess första tillämpning.

Även om kalmanfiltret är anpassat för linjära dynamiska system, så kan det även användas då systemet är olinjärt. Detta kräver dock beräkning av vissa statistiska storheter, såsom väntevärden och kovariansmatriser. Olika lösningar till dessa delproblem ger upphov till en rad olinjära varianter av kalmanfiltret. I denna avhandling ges en översikt av olika beräkningsmetoder för de nödvändiga statistiska storheterna, samt en jämförelse av dess fördelar och nackdelar. Flera nya insikter, samband och resultat ges.

Kalmanfiltret är det optimala linjära filtret, även om brussignalerna i ett linjärt system inte är normalfördelade. Däremot kan det finnas olinjära filter som är bättre. Om det förekommer stora avvikande värden på brussignalerna (s.k. outliers) så får kalmanfiltret stora avvikelser i sina skattningar som tar tid att bli av med. I denna avhandling undersöks därför students t-fördelningen, som kan modellera outliers. Ett filter som baseras på denna fördelning härleds. Den nya algoritmen är olinjär och övertygar vid ett mer utmanade scenario där vi jämför filtret med existerande algoritmer i litteraturen.

Acknowledgments

First, I wish to acknowledge my supervisor Fredrik Gustafsson. Fredrik keeps track of what his students are working on, encourages new ideas, and gives immediate feedback upon request — usually via e-mail around midnight. While being amazingly efficient, he still understands the challenges of being a parent.

I would also like to thank Svante Gunnarsson as representative and head of the automatic control group for accepting me as PhD student. Svante does a very good job in leading the control group, yet maintains a down-to-earth and friendly attitude towards all group members. Furthermore, Mikael Norrlöf is acknowledged for replying to my e-mail inquiry in early 2010, which eventually led to my employment in Linköping.

The group coordinators Åsa Karmelind (former) and Ninna Stensgård (current) have helped with all administrative issues. Furthermore, Åsa arranged a lovely apartment for my family and me. Thank you both!

Being a PhD student in the automatic control group means being well advised. I would like to thank Emre Özkan for his positive attitude towards research and confidence in my problem solving capabilities. Gustaf Hendeby has always shown interest in what I have been working on, which is very much appreciated. Additionally, I would like to thank Umut Orguner and Saikat Saha for their ideas and input.

Since I started in 2010, I have always had good company in my office. Therefore, I would like to acknowledge my office mates Henrik Ohlsson and Johan Dahlin. While Henrik had a tendency to drag large amounts of sand into the office, Johan keeps his half of the room rather tidy. Sometimes I wonder if he is not a bit more German than I am.

The first winter in Sweden was immensely brightened up by welcoming new friends. Tohid Ardeshiri is a very considerate person, a generous host, and has an amazing family. Niklas Wahlström does not only impress with his optimization skills: he can whistle like no one else can, and does not mind switching between Swedish, English, and German. Both never cease to give a helping hand.

I think that being a PhD student is sometimes challenging. After all, it is not easy to find a good trade-off between research, teaching, graduate courses, and life beyond university. All the more important is a good atmosphere among fellow PhD students. Expressed in the words of writer Mark Vonnegut,

> "*We are here to help each other get through this thing, whatever it is.*"

Therefore, I would like to acknowledge the PhD students in the group, some of which already graduated. In addition to the already mentioned, these are Christian Andersson Naesseth; Daniel Ankelhed; Patrik Axelsson; Jonas Callmer; André Carvalho Bittencourt, for the hint with the blue ink; Niclas Evestedt; Karl Granström; Ylva Jung; Sina Khoshfetrat Pakazad, for still asking me to join for a beer after my frequent declines; Manon Kok; Roger Larsson; Jonas Linder;

Fredrik Lindsten; Christian Lundquist; Christian Lyzell, for quality TikZ figures that I will include in the next thesis; George Mathai; Isak Nielsen; Hanna Nyqvist; Daniel Petersson; Peter Rosander; Maryam Sadeghi; Daniel Simon; Zoran Sjanic; Per Skoglar; Marek Syldatk, for sharing the enjoyable experience, together with Ylva and Daniel P., of writing a thesis in summer; Martin Skoglund; Lubos Vaci; Klas Veibeck; Johanna Wallén.

Gustaf, Fredrik, Emre, Manon, Niklas, Johan, and Tohid are acknowledged for their valuable comments that have significantly improved the manuscript. Niklas and Fredrik helped out with the Swedish summary. Furthermore, I would like to thank Gustaf and Henrik Tidefelt for developing and David Törnqvist for maintaining the thesis template.

I am lucky enough to have a number of very good friends in Linköping, Berlin, Braunschweig, and selected other places. Even though we do not get to see each other that often nowadays, I always look forward to the next meeting.

My parents Ingrid and Christian Roth, and my brothers Thomas and Andreas have always supported me in several ways. It would not have been easier without you, and I always enjoy being back at home. Also the Neupert family is gratefully acknowledged.

Finally, I want to thank my amazing wife Kora Neupert. Without your love and constant support, I would not have been able to finish this thesis. We have the privilege of watching Clara grow up, and that is something that I would not want to share with anyone else. Clara is gratefully acknowledged for making me smile everyday, and for the deep conversations that we share over breakfast. Over the last few weeks I could only spend little time with both of you, and I hope I can make up for that soon.

Financial support for this thesis has been provided by the MC Impulse project, a European commission FP7 Marie Curie initial training network under grant agreement number 238710.

Linköping, Sweden Michael Roth

Contents

1 Introduction

The technical program of any larger signal processing conference, for instance the *International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), overwhelms the attendee with a countless number of applications, specific problems, and suggested solutions how to handle these. Although each paper is individually very specific, most of the publications at such a conference follow one common theme. Namely, the researchers want to *extract useful information* from available *data*.

Motivated by the aforementioned common structure in many signal processing problems, the focus of this thesis is on algorithms rather than applications. We address *filtering algorithms* for *state space models* that are *nonlinear* and/or subject to *heavy-tailed noise*. Here, filtering refers to processing an incoming stream of measurements online, in order to calculate up-to-date estimates of the *system states*. Hence, the filtering problem is one instance of a *state estimation problem*, and a very general scheme into which many real world signal processing applications can be cast.

We introduce the following example to acquaint the reader with such a filtering problem.

Example 1.1: Air traffic control

A crucial requirement for air traffic control around highly frequented airports is the knowledge of the positions and the velocities of all surrounding aircraft.

We first consider one individual aircraft. Its position and velocity can be described by a six-dimensional vector that changes over time. This vector subsumes all relevant information for the staff at the airport tower: it is the system state that we would like to estimate.

The aircraft motion is determined by physical laws and as such lends itself to deriving a motion model. This model describes how the state changes over time. Obviously, the thrust and steering actions that the pilot executes must be taken into account here. These are not known to the staff at the airport. However, the staff know the typical motion that civilian aircraft usually perform. In the model, such knowledge can be included as random input with a certain probability distribution. In the state estimation problem, such random inputs are called process noise. In summary, each aircraft has a dynamic model which describes its motion in a probabilistic manner.

The airport tower is equipped with radar sensors that provide information about the distance to individual aircraft and the angles in which they are approaching. Such measurements are taken at specific times and offer only a snapshot of the changing distances and angles. Furthermore, they are subject to inevitable measurement errors which are in the filtering context accounted for by another random variable with a certain probability distribution: the measurement noise. Using simple geometry, the range and angle measurements can be related to the three-dimensional aircraft position, that is: to some components of the state vector. The functional relation between measurements and states, together with the measurement noise, yields a so called measurement model. Here, the measurement model is nonlinear because the range and angles have a nonlinear relation to the Cartesian position vector.

For one aircraft, the filtering problem amounts to estimating its state based on the incoming radar measurements and the dynamic and measurement model. The filtering is typically carried out online, which means that obtained measurements are immediately processed to give an up-to-date state estimate.

The above example is in several ways instructive.

First, it highlights that many complex real world applications contain elementary state estimation problems. The above air traffic control example is actually an instance of a *multiple target tracking* problem which entails further difficulties. Conceptually similar is the problem of tracking obstacles in the vicinity of a driverless car, which emphasizes the practical relevance of tracking for future technology. The solutions to target tracking problems often rely on running one filter for each subsystem or target, and subsequently combining the results in a clever way. Hence, filtering algorithms are an essential ingredient in solving target tracking problems

Second, the state space models can involve nonlinear dynamic or measurement relations. In the above example, the radar yields a nonlinear measurement model. This prohibits the problem to be solved exactly, as the commonly accepted optimal solution cannot be computed. In fact, the optimal *Bayesian filtering recursions* can only be computed for very few specific problems, as we shall see. Even if the dynamic and measurement models are linear, the noise distributions might complicate things further. Sensors might malfunction at certain times and provide meaningless values, or *clutter*. Also, the dynamic model might be driven by input signals that exhibit rapid changes. In the aircraft example, for instance, a pilot might decide to take an unusual dive. Such phenomena can be modeled by *heavy-tailed* process and measurement noise distributions that, however, prohibit tractable solutions to the Bayesian filtering recursions. The lack of exact solutions calls for the approximate filtering algorithms that are considered in this thesis.

From an algorithmic point of view, the actual application is secondary. As long as a state space description of the considered system or process is provided, the algorithms that we present in this thesis can be employed to gather valuable state information from noisy measurements. Applications are thus found in as diverse areas as *biomedical signal processing*, *chemical process control*, *economics*, or *field robotics*, just to name a few.

The thesis should be accessible by readers with a knowledge of basic probability theory and state space models. This hopefully includes a number of practitioners. Although large parts of this thesis are self-contained, the format of a licentiate thesis requires a certain brevity. Therefore, the reader is consistently referred to sources from extensive bibliography.

1.1 Considered Problems

The filtering problem is in general difficult, especially if the underlying system is nonlinear and/or the noise sources are non-Gaussian. We approach it using relatively simple algorithms that are in the spirit of the *Kalman filter* and aim at computing the mean and covariance matrix of the filtered state. The simplicity of the algorithms comes at the price that they certainly do not manage to solve every estimation problem. If, however, the true underlying state distribution is close to Gaussian, that is: roughly bell-shaped and symmetric about its mean, the algorithms that we present have a good chance of success. Beyond the Gaussian distribution, we suggest to use the t distribution in filtering problems with heavy-tailed noise and develop a filter that employs this idea. Finally, all of the algorithms in this thesis can be used as building blocks in more advanced schemes such as filter banks or (marginalized) particle filters.

Delving more into details, the thesis considers two core problems that are covered in dedicated chapters.

Computing expected values of nonlinearly transformed random variables

The problem of computing *expected values* is an essential requirement to run the *Kalman filter* for *nonlinear systems*, but appears also in other problems such as conversion of statistical information from one coordinate frame to another. Given extensive computational resources, expected values can be computed using Monte Carlo methods. However, in online applications (for example to compute expected values in a Kalman filter) and/or high-dimensional spaces, faster alternatives are desired. We therefore investigate methods that are based on deterministic sampling, which include the *unscented transformation* and *deterministic integration rules*. Furthermore, methods that rely on simplifying the original problem are considered, including *truncated Taylor series* and *interpolation approaches*. All of the methods are approximate in general but exact for certain problem instances.

Filtering in linear systems with heavy-tailed noise

From an application perspective, heavy-tailed noise distributions can be used to model phenomena such as measurement outliers from unreliable sensors, sudden changes in unknown input signals and maneuvers in tracking applications, and effects that are caused by simplified modeling of complex real world processes.

Linear systems that are subject to *heavy-tailed process and measurement noise* are in general difficult to handle. We derive a filter that is based on Student's t distribution, which is known to have heavier tails than the Gaussian, and thereby address the outlier problem in a natural manner. The resulting algorithm can be viewed as an *assumed density filter* because the state and measurement are assumed jointly t distributed. The recursive formulas exhibit some similarities with the Kalman filter but provide the important feature that the measurement affects also the involved matrix calculations, in contrast to the Kalman filter in which the covariance matrices are independent of the measurement.

1.2 Contributions

We here list the main contributions of the thesis in the order of appearance.

Chapters 2 and 3 are directed towards a general signal processing audience.

- A comprehensive introduction to state estimation problems is given. Common solutions are surveyed, including the Kalman filter, nonlinear Kalman filter variants, and the particle filter.
- We emphasize that the essential step in any nonlinear Kalman filter is the computation of a number of mean values and covariance matrices. These are all expected values of nonlinearly transformed random variables.

Chapter 4 is focused on computing expected values. This clearly relates to nonlinear Kalman filter algorithms, but is a more general problem.

- An extensive collection of easy to understand and easy to implement moment computation schemes is provided.
- Elliptically contoured distributions are discussed. This facilitates extending common knowledge about the Gaussian distribution to a larger family of distributions. We show that certain n-dimensional expected values can be reduced to scalar problems.
- A compact re-formulation of the multidimensional Taylor series facilitates computing moments up to arbitrary precision, and gives rise to new investigation methods for nonlinear models.
- Computer algebra systems and symbolic computations are re-considered to extend their use in the MATLAB dominated filtering community.

- The special case of quadratic functions with Gaussian input is thoroughly investigated and used to derive a sampling based implementation of the moment expressions.
- We give accuracy statements for each of the methods, and conclude that sometimes very little can be said.
- The methods are compared on two instructive examples.

Chapter 5 and Appendix A.3 discuss Student's *t* distribution and its use in filtering problems.

- An extensive appendix on the *t* distribution is provided, including a comparison with the Gaussian distribution.
- A filtering algorithm that can handle heavy-tailed process and measurement noise, and is entirely based on the *t* distribution, is developed.
- The new filter is compared with state-of-the-art algorithms on a challenging tracking example.

1.3 Publications

Parts of the material in this thesis have been published by the author.

The nonlinear filters of Section 3.2 and specifically the scheme that we present in Section 4.6 relate to

> M. Roth and F. Gustafsson. An efficient implementation of the second order extended Kalman filter. In *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, July 2011.

The particle filter example of Section 3.3 is taken from

> M. Roth, F. Gustafsson, and U. Orguner. On-road trajectory generation from GPS data: A particle filtering/smoothing application. In *15th International Conference on Information Fusion (FUSION)*, July 2012.

The filter development of Chapter 5 is covered in

> M. Roth, E. Özkan, and F. Gustafsson. A Student's *t* filter for heavy tailed process and measurement noise. In *38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

The material in Appendix A.3 has been published as technical report

> M. Roth. On the multivariate *t* distribution. Technical Report LiTH-ISY-R-3059, Automatic Control, Linköping University, Linköping, Sweden, Apr. 2013.

2 State Estimation

This chapter sets the stage for all following chapters by introducing the central problem of this thesis: estimating the unknown states of a dynamical system. First, a general introduction to stochastic state space models is given in Section 2.1. The qualifier stochastic refers to the presence of process noise, measurement noise, and an uncertain initial state. Section 2.2 defines three different types of state estimation problems, including the filtering problem that is central to this thesis. Recursive formulas for filtering densities are provided in Section 2.3, and followed by a brief discussion on computing point estimates from the estimated densities. The chapter is concluded by a brief treatment of state variable changes in Section 2.5.

2.1 Stochastic State Space Models

This section builds upon (deterministic) state space models as treated in a first course in automatic control. Appropriate introductions can be found in, for instance, Gustafsson (2010b) or Bar-Shalom et al. (2001).

A core concept in dynamical systems is the state x: a quantity that captures relevant information about the system and typically changes over time. The way in which the state evolves is governed by a number of equations that either describe x as a function of time t (by means of differential equations) or at discrete-time instances k (by means of difference equations). The state itself, although of core interest, is in general not available directly. Instead, measurements y that relate to the state are taken at specific times. This section explains these ideas in detail and provides the required formalism.

Continuous-time state space models appear naturally when a real world process is modeled using physical reasoning, because the state dynamics are often described by a number of coupled differential equations. Despite this, we restrict ourselves to discrete-time descriptions. These are, of course, often obtained by discretization of continuous-time models. Difference equations appear instead of differential equations and provide extra modeling flexibility as we shall see. Moreover, the restriction to discrete-time systems facilitates straightforward implementation and a much simpler treatment of stochastic signals, which would otherwise require extra care and the use of stochastic differential equations as treated in Jazwinski (1970).

General model

In a very generic form, a stochastic state space model consists of the equations

$$x_{k+1} = f_k(x_k, v_k), \tag{2.1a}$$

$$y_k = h_k(x_k, e_k). \tag{2.1b}$$

The state difference equation (2.1a) relates the upcoming state x_{k+1} to the current state x_k and the process noise v_k, via a function f_k. The measurement equation (2.1b) relates x_k and the measurement noise e_k to the observation y_k, via a function h_k. All of the quantities x_k, v_k, y_k, and e_k are vectors with known dimensions, for instance x_k and y_k have respective dimensions n_x and n_y.

Furthermore, a probabilistic description of the noise signals v_k and e_k, and the initial state x_0 has to be specified in terms of a joint probability distribution. Often, marginal probability densities, as in

$$v_k \sim p(v_k), \quad e_k \sim p(e_k), \quad x_0 \sim p(x_0), \tag{2.1c}$$

are given with some additional statements that regard dependence or correlation between v_k, e_k, and x_0. The compact notation in (2.1c) means that v_k, for instance, has a distribution with density $p(v_k)$.

The following assumptions are made for the problems that we treat in this thesis. The initial time index is $k = 0$ and the first measurement is taken at $k = 1$. Both functions f_k and h_k are known for each time instance k. The noise characteristics of v_k and e_k are known, and so is the distribution of x_0. We assume v_k and e_k to be white for the sake of simplicity, that is $\operatorname{cov}\{v_k, v_l\} = 0$ for $k \neq l$. The initial state x_0 is assumed uncorrelated to e_k and v_k.

The state space model (2.1) is general in several aspects. First, the involved signals and the state in particular are not restricted to be continuous but could assume discrete values from a finite set. Second, the functions f_k and h_k can contain arbitrary nonlinearities. The subscript k indicates that both might be time-varying, which allows for including deterministic input signals as in many control applications. Finally, the involved noise signals v_k and e_k, and the initial state can admit arbitrary distributions. For notational convenience, we assume that the density functions in (2.1c) exist.

General model with additive noise

A widely used special case of (2.1) is the stochastic state space model with additive noise:

$$x_{k+1} = f_k(x_k) + v_k, \tag{2.2a}$$
$$y_k = h_k(x_k) + e_k. \tag{2.2b}$$

Here, v_k and e_k are often assumed independent. Then, an alternative description of the model in terms of conditional densities $p(x_{k+1}|x_k)$ and $p(y_k|x_k)$ is easily derived. A number of approximate filtering techniques given in Section 3.2 rely on (2.2) with a Gaussian assumption on v_k and e_k. The related algorithms boil down to approximately computing expected values with respect to Gaussian densities, a problem that we address in Chapter 4.

Sometimes v_k is pre-multiplied by a matrix G_k, which allows for process noise with $n_v < n_x$. The product $G_k v_k$ is then partially deterministic and has a covariance matrix that is merely positive semi-definite. Although a similar matrix factor is conceivable for the measurement equation, it is seldom seen. One reason is that all measurements are assumed to be noise-corrupted, which requires the covariance of e_k to be positive definite. For such a case $n_y = n_e$ and the measurement noise can be fully characterized by $p(e_k)$ without the need for an extra factor.

Linear model

Most common in literature, well researched and well understood, is the linear state space model

$$x_{k+1} = F_k x_k + v_k, \tag{2.3a}$$
$$y_k = H_k x_k + e_k, \tag{2.3b}$$

which has been at the core of control and estimation theory for several decades. For a discussion of extra matrices that accompany v_k and e_k, the reader is referred to the preceding paragraphs.

For the model described by (2.3) with arbitrary noise distributions with known first and second moments, there exists an optimal unbiased state estimation algorithm that provides the lowest mean squared estimation error: the celebrated Kalman filter that we introduce in Section 3.1. For Gaussian densities in (2.1c), x_k and y_k are Gaussian at any time k, and the resulting *filtering recursions* are solved using, again, the Kalman filter. It should be noted that (2.3) could be obtained by linearization of (2.1) or (2.2), an idea which is exploited in a family of approximate filtering algorithms including the *extended Kalman filter* of Section 3.2.

2.2 Bayesian Filtering, Smoothing, Prediction

State estimation in stochastic state space models amounts to gathering information about the random variable x_k in a probabilistic fashion. Therefore, it is reasonable to aim at recovering the probability density function of x_k given a stream

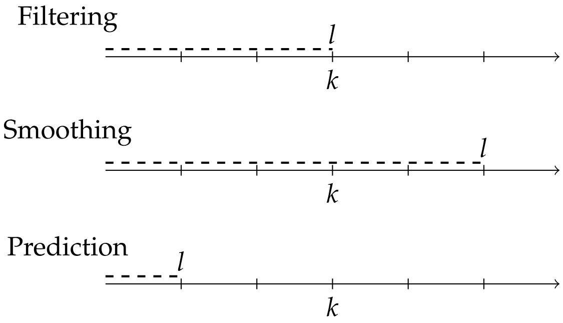


Figure 2.1: Schematic illustration of the different estimation problems. The user is interested in the state at time k. The available measurements up to time l are illustrated by the dashed line.

of related random variables, the measurements $y_{1:l} = \{y_1, \dots, y_l\}$. The interpretation of the state as a random variable with a certain distribution is in accordance with the *Bayesian* viewpoint, and the sought after conditional density $p(x_k|y_{1:l})$ is often called the *posterior density*. Unfortunately, the ambitious intention to calculate the posterior is in most cases computationally intractable because it cannot, in general, be described by a finite number of parameters. Nevertheless, we continue to describe the conceptual estimation problems and postpone a discussion of actual algorithms to Chapter 3.

Depending on the temporal relation between the time of interest k and the time interval of available observations, determined by l, three different estimation problems can be distinguished. These are also covered in Bar-Shalom et al. (2001).

Filtering

In a *Bayesian filtering* problem $l = k$, that is all measurements up to time k are available for (conceptually) calculating $p(x_k|y_{1:k})$. The first row in Figure 2.1 illustrates this scenario. The dashed line symbolizes the set of available measurements which reaches up to and includes time k. Clearly, this is an online problem in the following sense: as soon as a measurement y_k becomes available, we can compute the *filtering density* $p(x_k|y_{1:k})$.

Smoothing

In a *Bayesian smoothing* (or retrodiction) problem $l > k$ and measurements beyond k are available to compute the posterior. This is illustrated in the second row of Figure 2.1. Intuitively, the look-ahead should result in an improvement over filtering. The density $p(x_k|y_{1:l})$ is called *marginal smoothing density*, in contrast to the *joint smoothing density* of all states $p(x_{0:l}|y_{1:l})$. Marginal smoothing problems can be further divided. First, there is *fixed-lag* smoothing where both k and $l = k + m$ increase over time. Here, m is called lag and a positive integer that determines the look-ahead horizon. This yields an online scheme for estimating past states employing current measurements. In *fixed-interval* smoothing, in contrast, the set of available measurements is fixed and k varies from 0 to l. That is, all measurements need to be collected in advance and the computations are carried out offline. Finally, in *fixed-point* smoothing k is fixed and $p(x_k|y_{1:l})$ is updated as more and more measurements become available.

Prediction

The third row in Figure 2.1 illustrates a *prediction* scheme in which $l < k$. The available measurements $y_{1:l}$ are used to compute a *prediction density* of state in the future. There are similarities to the marginal smoothing problem, and also prediction can be divided into three categories (with *fixed-lead* prediction as analogue to fixed-lag smoothing). Further details can be found in Bar-Shalom et al. (2001).

2.3 Recursive Formulas for the Filtering Density

After having defined the Bayesian filtering problem in the previous section, we now dig further into the conceptual solution of recursively updating the probability density $p(x_k|y_{1:k})$ as k increases. We present two different derivations of these *filtering recursions*. Most of the material can be found in as early references as Jazwinski (1970).

2.3.1 Markov Property and Conditional Independence

State space models with independent white noise impose a *Markov property* on the state, that we here briefly explain. Suppose that we want to simulate one state transition (2.1a) for a given v_k, and suppose that we have access to all previous states $x_{0:k} = \{x_0, \ldots, x_k\}$. It is obvious from (2.1a) that only x_k is required to compute x_{k+1}. In other words, x_{k+1} is conditionally independent of $x_{0:k-1}$ given x_k, and we say that the state is a *Markov process* with the conditional density

$$p(x_{k+1}|x_{0:k}, y_{1:k}) = p(x_{k+1}|x_k). \tag{2.4}$$

Reasoning in a similar manner for the measurement equation (2.1b), we can formulate

$$p(y_k|x_k, y_{1:k-1}) = p(y_k|x_k). \tag{2.5}$$

Employing tools that are described in Bishop (2006), a *graphical model* can be used to illustrate the probabilistic dependencies between state and measurements for consecutive time instances. Figure 2.2 contains such a graphical model from which the statements (2.4) and (2.5) can be recovered as well.

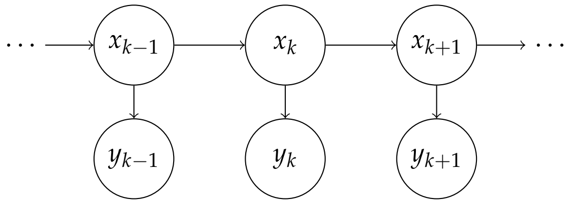


Figure 2.2: *Graphical model for a state space model.*

2.3.2 Filtering Recursions I: Prior and Likelihood

An alternative to specifying the state and measurement equation of (2.1) is the description

$$x_{k+1} \sim p(x_{k+1}|x_k), \tag{2.6a}$$
$$y_k \sim p(y_k|x_k), \tag{2.6b}$$

which makes use of conditional probability densities to specify the ways in which the state can evolve. For the state space model (2.2) with additive and independent noise sources, (2.6) can be easily recovered:

$$p(x_{k+1}|x_k) = p_{v_k}\Big(x_{k+1} - f_k(x_k)\Big), \tag{2.7a}$$
$$p(y_k|x_k) = p_{e_k}\Big(y_k - h_k(x_k)\Big), \tag{2.7b}$$

where p_{v_k} and p_{e_k} denote the densities that determine the process and measurement noise, respectively.

We now turn to the filtering posterior again and gradually take it apart:

$$\begin{aligned} p(x_k|y_{1:k}) &= \frac{p(x_k, y_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} && \text{(2.8a)} \\ &= \frac{p(y_k|x_k, y_{1:k-1})p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} && \text{(2.8b)} \\ &= \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}. && \text{(2.8c)} \end{aligned}$$

(2.8a) is obtained via Bayes' theorem, and (2.8b) using the rules for joint probability densities. Finally, (2.8c) is obtained using the conditional independence result (2.5). We have recovered $p(y_k|x_k)$ of (2.6b), which is in this context referred to as the *likelihood* of x_k.

The factor $p(x_k|y_{1:k-1})$ in the numerator of (2.8c) is termed the *prior density* of x_k, and obtained by the following integration:

$$\begin{aligned} p(x_k|y_{1:k-1}) &= \int p(x_k, x_{k-1}|y_{1:k-1})\,\mathrm{d}x_{k-1} && \text{(2.9a)} \\ &= \int p(x_k|x_{k-1}, y_{1:k-1}) p(x_{k-1}|y_{1:k-1})\,\mathrm{d}x_{k-1} && \text{(2.9b)} \\ &= \int p(x_k|x_{k-1}) p(x_{k-1}|y_{1:k-1})\,\mathrm{d}x_{k-1}. && \text{(2.9c)} \end{aligned}$$

The idea is to marginalize out the previous state from the joint density. The result (2.9c) is then obtained using the conditional independence result (2.4) and involves the posterior at the previous time step, $p(x_{k-1}|y_{1:k-1})$.

The denominator in (2.8c) is a normalization constant that does not depend on x_k. It can be computed using the integral

$$p(y_k|y_{1:k-1}) = \int p(y_k|x_k) p(x_k|y_{1:k-1})\,\mathrm{d}x_k \tag{2.10}$$

and is sometimes referred to as *marginal likelihood* or *evidence*.

Loosely speaking, the posterior is determined by the product of likelihood and prior. This set-up is the very basis of much of *Bayesian statistics*, and frequently encountered in the corresponding literature. The reader is referred to Bishop (2006) or DeGroot (2004) for the concept, and Gustafsson (2010b) for its application in a state estimation setting.

2.3.3 Filtering Recursions II: Transformation Theorem

This section presents an alternative derivation of the filtering recursions, which is based on the *transformation theorem* of Appendix A.1. The calculations are based on manipulation of probability density functions and apply to the general state space model (2.1).

Consider the model (2.1) with the following joint density of state and noise signals

$$\begin{bmatrix} x_{k-1} \\ v_{k-1} \\ e_k \end{bmatrix} \sim p(x_{k-1}, v_{k-1}, e_k|y_{1:k-1}) := p_{xve}\Big(\begin{bmatrix} x_{k-1} \\ v_{k-1} \\ e_k \end{bmatrix}\Big). \tag{2.11}$$

Such a joint density is useful in cases where $(x_{k-1}, v_{k-1}, e_k)|y_{1:k-1}$ are uncorrelated but not independent. One example is treated in Chapter 5 where Student's t distribution is employed for (2.11).

We next utilize the transformation theorem, and an auxiliary variable, to obtain the joint density of x_k and y_k given $y_{1:k-1}$. Consider z_k such that there exists a one-to-one relation between x_{k-1}, v_{k-1}, e_k and x_k, y_k, z_k. This might not always be achievable, but we show how z_k can be chosen for a commonly encountered case. Hence, we have available the mapping

$$\begin{bmatrix} x_{k-1} \\ v_{k-1} \\ e_k \end{bmatrix} = \phi(x_k, y_k, z_k). \tag{2.12}$$

Straightforward application of the transformation theorem now gives

$$p(x_k, y_k, z_k | y_{1:k-1}) = p_{xve}\Big(\phi(x_k, y_k, z_k)\Big) | \det(J) |, \tag{2.13}$$

where J is the Jacobian of $\phi(x_k, y_k, z_k)$ and contains first-order partial derivatives with respect to all components in x_k, y_k, and z_k.

The joint density can be obtained by integrating out z_k

$$p(x_k, y_k | y_{1:k-1}) = \int p(x_k, y_k, z_k | y_{1:k-1}) \, \mathrm{d}z_k. \tag{2.14}$$

The filtering posterior is then given by a conditioning step, or Bayes' law, that involves similar integral calculations to the ones in the previous section

$$p(x_k | y_{1:k}) = \frac{p(x_k, y_k | y_{1:k-1})}{p(y_k | y_{1:k-1})}. \tag{2.15}$$

It is obvious that although we can find conceptual solutions in terms of integral equations, these lack analytical or tractable solutions in most cases.

Example 2.1: Linear system

This example is about finding the filtering recursions in a linear system by employing the transformation theorem. The state and measurement equations (2.3) are re-stated for convenience

$$x_{k+1} = F x_k + v_k, \tag{2.16a}$$
$$y_k = H x_k + e_k. \tag{2.16b}$$

Using the auxiliary variable $z_k = x_{k-1}$, we can formulate a linear relation

$$\begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} = \begin{bmatrix} F & I & 0 \\ HF & H & I \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ v_{k-1} \\ e_k \end{bmatrix}, \tag{2.17}$$

which in turn implies a mapping of the form (2.12):

$$\begin{bmatrix} x_{k-1} \\ v_{k-1} \\ e_k \end{bmatrix} = \begin{bmatrix} 0 & 0 & I \\ I & 0 & -F \\ -H & I & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix}. \tag{2.18}$$

Here, ϕ is linear and its Jacobian corresponds to the matrix in (2.18). The related Jacobian determinant can be readily computed using block matrix formulas and is one.

Adopting the notation of (2.11) and using $z_k = x_{k-1}$ yields

$$p(x_k, y_k, x_{k-1}|y_{1:k-1}) = p_{xve}\Big(\begin{bmatrix} x_{k-1} \\ x_k - Fx_{k-1} \\ y_k - Hx_{k-1} \end{bmatrix}\Big). \tag{2.19}$$

Next, marginalization of x_{k-1} and conditioning on y_k are required. Whether these can be computed in a simple manner depends on p_{xve}, which represents the joint distribution of $(x_{k-1}, v_{k-1}, e_k)|y_{1:k-1}$. For a joint t distribution, for instance, marginalization is simple as shown in Appendix A.3. Also, then the conditioning is possible. Completing the details yields the filtering algorithm of Chapter 5. Matters simplify further if $(x_{k-1}, v_{k-1}, e_k)|y_{1:k-1}$ are jointly Gaussian and uncorrelated because then the variables are also independent and p_{xve} factors into $p(x_{k-1}|y_{1:k-1})p(v_{k-1})p(e_k)$.

2.3.4 The Lack of a Tractable Solution

Although we derived a number of compact integral equations that describe the filtering density recursively, the actual calculations cannot be executed except for very few special cases. One reason for this lies in the integrals that we need to calculate when marginalizing, as in (2.9c) or (2.10). These quickly turn out to be daunting for nonlinear systems. But even if the integrals can be computed, there might be other difficulties that relate to the number of parameters that are needed to describe the posterior $p(x_k|y_{1:k})$. We explain this using a simple linear example.

Example 2.2: Gaussian mixture noise

We consider the following system in which the state is available as noisy measurement:

$$x_{k+1} = x_k + v_k, \tag{2.20a}$$

$$y_k = x_k + e_k. \tag{2.20b}$$

For the sake of simplicity, we assume that all signals are scalar. The initial state and process noise are mutually independent and Gaussian, with densities $p(x_0) = \mathcal{N}(x_0; \hat{x}_0, P_0)$ and $p(v_k) = \mathcal{N}(v_k; 0, Q)$. The measurement noise admits a mixture distribution with two Gaussian components. Its density is given by

$$p(e_k) = \frac{1}{2}\mathcal{N}(e_k; 0, R_1) + \frac{1}{2}\mathcal{N}(e_k; 0, R_2). \tag{2.21}$$

The simple structure of $p(x_0)$, $p(v_k)$, and $p(e_k)$ allows for direct application of the Bayesian filtering recursions of Section 2.3.2. Starting with a unimodal $p(x_0)$, we obtain $p(x_1|y_1)$ as mixture of two and $p(x_2|y_{1:2})$ as mixture of four Gaussian densities. Obviously, the number of mixture components N_k in $p(x_k|y_{1:k})$ increases exponentially as $N_k = 2^k$.

Now, a scalar Gaussian mixture density with N_k components requires N_k mean values, N_k variances, and $N_k - 1$ weights to be fully specified. Accordingly, the required number of scalar parameters to describe, for instance, $p(x_{10}|y_{1:10})$ exactly is 3071. Even in this scalar example, the sheer number of parameters to describe the exact posterior becomes prohibitive after few time steps.

2.4 State Estimates from the Filtering Density

Within the Bayesian framework, we specified the filtering problem in terms of finding a conditional probability density $p(x_k|y_{1:k})$, the filtering posterior. The user, however, is in general interested in a point estimate $\hat{x}_{k|k}$ of the state and some indication of its confidence, often represented by a covariance matrix $P_{k|k}$. Here, $\hat{x}_{k|k}$ is shorthand notation for the state estimate at time k given all measurements from time 1 to k.

The predominant idea behind point estimate computations is minimization of some expected loss, where the expectation is carried out with respect to the posterior density. An interesting discussion can be found in Anderson and Moore (1979), where it is shown that the conditional mean $\mathrm{E}\{x_k|y_{1:k}\}$ minimizes the expected squared estimation error, regardless of the shape of $p(x_k|y_{1:k})$. The conditional mean is therefore also known as *minimum mean squared error* or MMSE estimate. For a one-dimensional x_k, it can be shown that the minimum expected absolute estimation error is achieved for $\hat{x}_{k|k}$ as *median* of $p(x_k|y_{1:k})$. Regardless of the state dimension, $\hat{x}_{k|k}$ can alternatively be chosen as the point that maximizes the posterior density, the posterior *mode*. Then $\hat{x}_{k|k}$ is known as *maximum a posteriori* or MAP estimate.

For linear Gaussian systems, the filtering density is Gaussian throughout time and as such fully described by an estimate and its covariance. In that respect, there is no difference between computing the full posterior $\mathcal{N}(x_k; \hat{x}_{k|k}, P_{k|k})$ and merely extracting $\hat{x}_{k|k}$ and $P_{k|k}$. In fact, $\hat{x}_{k|k}$ and $P_{k|k}$ are what is called *sufficient statistics* for the Gaussian distribution. Furthermore, $\hat{x}_{k|k}$ unifies mean, mode, and median for a Gaussian filtering density $\mathcal{N}(x_k; \hat{x}_{k|k}, P_{k|k})$. The resulting filtering algorithm is nothing but the ubiquitous Kalman filter of Section 3.1.

Instead of attempting to estimate the full posterior density and extracting an estimate in a second step, most algorithms focus on computing an estimate directly. In general, such techniques accompany approximations of one sort or another. Common approaches include the nonlinear Kalman filter variants that we introduce in Section 3.2.

2.5 Change of State Variables

We conclude this chapter with some remarks on how the state coordinate frame can be changed. In light of the previously derived recursions for the filtering posterior, we know that exact calculations might not be possible. Therefore, one needs to resort to approximations of the filtering density. This is where state variable changes can be applied to turn a difficult problem into a hopefully simpler one that provides the user with equally useful information. A change of state variables can, for instance, be used to transform a system that contains nonlinearities in state and measurement equation into a different system that has linear state or measurement equations.

Example 2.3: Tracking in different coordinate frames

We consider a vehicle that travels in a two-dimensional plane. Its motion appears to be driven by a random velocity input. We would like to estimate the position but the only available measurements are range and bearing provided by some noisy sensor.

The state vector for this simple example consists of the two-dimensional position that we would like to estimate. The measurement model can be written as

$$y_k = \begin{bmatrix} r_k \\ \phi_k \end{bmatrix} + e_k = \begin{bmatrix} \sqrt{x_{1,k}^2 + x_{2,k}^2} \\ \arctan\left(\frac{x_{2,k}}{x_{1,k}}\right) \end{bmatrix} + e_k \tag{2.22}$$

with the position x_1 and x_2 and the measurement noise e_k. Regardless of the state dynamics that we assume, the measurements are always nonlinear in Cartesian coordinates x_1 and x_2, and an approximate nonlinear filter must be employed for the filtering task.

Now, if the user is able to process the position in *polar* coordinates r and ϕ as well, range and bearing can be chosen as state variables instead. The measurement model is then linear. Of course, the state dynamics have to be changed accordingly, which can lead to introduction of nonlinearities in the state equation.

Such examples become relevant in the field of *target tracking*, as treated in Blackman and Popoli (1999).

We next generalize the above example. Apart from obtaining "less nonlinear" models, a change in state coordinates might be motivated by observability issues. Some states might be harder to estimate than others, and it is sometimes useful to split the state vector into parts that are more challenging and less challenging to observe. One such example is given by target tracking in *modified spherical coordinates*, as performed in Mallick et al. (2012).

The results that we present are general in the sense that we treat linear as well as nonlinear variable changes, and linear as well as nonlinear systems. However, we restrict ourselves to static variable change relations and the discrete-time case. A more general treatment for linear systems and time-varying variable change relations, also in continuous-time, can be found in Rugh (1995).

Linear system, linear state transformation

We consider the discrete-time linear system described by (2.3) and assume that we have available an invertible matrix T such that $x_k = Tz_k$, regardless of the time instance k. Substituting all x in (2.3) and multiplying the state equation by T^{-1} yields the linear system

$$z_{k+1} = T^{-1}F_kTz_k + T^{-1}v_k, \tag{2.23a}$$

$$y_k = H_kTz_k + e_k \tag{2.23b}$$

with the state variable z. The state dynamics are determined by the matrix $T^{-1}F_kT$ and the linearly transformed process noise $T^{-1}v_k$. Similarly, the measurement equation in z has a matrix H_kT instead of H_k.

General system and transformation

The motivation for treating state variable changes in this thesis mainly stems from the application in nonlinear filtering tasks. Here, the user might be able to improve the estimation results by slightly altering the problem. A discussion on the use of variable changes in target tracking can be found in Blackman and Popoli (1999). Gustafsson and Isaksson (1996) compare different state coordinate frames for one specific tracking problem. Also, the book chapter Mallick et al. (2012) applies the technique that we describe.

Suppose that we have available a static relation between the state x and z, that is valid for all time instances k:

$$x_k = \tau(z_k), \quad z_k = \sigma(x_k). \tag{2.24}$$

Then, simple substitution of x in the state difference equation of the general model (2.1) yields

$$\tau(z_{k+1}) = f\Big(\tau(z_k), v_k\Big), \tag{2.25}$$

a difference equation in z. Continuing in a similar way as in the linear case, we obtain a general state space model with state variable z:

$$z_{k+1} = \sigma\Big(f(\tau(z_k), v_k)\Big), \tag{2.26a}$$

$$y_k = h\Big(\tau(z_k), e_k\Big). \tag{2.26b}$$

Changing the state coordinate frame goes along with a number of issues. Prior knowledge, for instance, might only be available in terms of the distribution of x_0. If we would like to run a Kalman filter in z-coordinates, then the mean and covariance matrix of z_0 must be computed from $p(x_0)$. This is one instance of a more general problem that we extensively treat in Chapter 4. A conceptually similar problem appears if we have computed an estimate and covariance for z_k and would like to convert these statistics back to x-coordinates.

3

Filtering Algorithms

In this chapter, we provide the reader with a number of tools to approach the filtering problem that we defined in the preceding chapter. In Section 3.1, the Kalman filter is introduced as an exact and optimal estimator for linear systems with known noise statistics. In Section 3.2, we introduce algorithms that build on the Kalman filter but consider nonlinear state space models. These filters are approximate in the sense that they do not capture the entire filtering density but merely aim at propagating a state estimate and its covariance matrix throughout time. Finally, we introduce the conceptually different particle filters which are based on an approximation to the entire filtering density in Section 3.3.

3.1 The Kalman Filter

We begin this chapter by introducing the ubiquitous Kalman filter, also called the "workhorse of estimation" according to Bar-Shalom et al. (2001).

3.1.1 Background

Kalman published his seminal paper on estimation in discrete-time state space models, Kalman (1960), in a mechanical rather than electrical engineering journal. This indicates that he expected his at the time unconventional ideas to be met with skepticism. After all, the field of electrical engineering was dominated by frequency domain approaches and the *Wiener filter*. Kalman deviated from this common trend and provided recursive formulas that were tailored for implementation. Furthermore, his framework allowed for non-stationary processes and in this respect went beyond the Wiener filter.

Although similar ideas were developed earlier in Swerling (1959), Kalman was fortunate to be just in time to provide the tools for solving a navigation problem in the NASA Apollo program. Interestingly, this first application involved a nonlinear problem and led to the immediate derivation of the *extended Kalman filter*. More extensive accounts of Kalman filter history are given in Kailath et al. (2000) and Grewal and Andrews (2010).

Kalman presented his filter first in Kalman (1960) for the discrete-time case, and later issued the continuous-time variant in Kalman and Bucy (1961). Some of the early classics on the topic, such as Jazwinski (1970) and Maybeck (1979, 1982), mainly focus on continuous-time dynamics. Anderson and Moore (1979) can be considered as early standard text with focus on discrete-time, and provides a number of alternative derivations and many extensions to the Kalman filter. Kailath et al. (2000) provides a very comprehensive treatment of linear estimation with, of course, the Kalman filter in its center. Here, relations to the Wiener filter are also covered. Verhaegen and Verdult (2007) assumes a least-squares perspective and establishes relations to the field of *system identification*. Gustafsson (2010b) and Bar-Shalom et al. (2001) treat application to *navigation* and *tracking*.

3.1.2 Basic Algorithm

The discussion in this section is restricted to linear systems.

The Kalman filter is an algorithm for estimating the state of linear state space systems described by (2.3). The filter can be derived as optimal estimator for several criteria, under several more or less restrictive assumptions. We will not reproduce all derivations here but instead refer the reader to the different approaches in Anderson and Moore (1979). The presentation in Kailath et al. (2000) is possibly closest to Kalman's seminal paper, which is reviewed by the authors in a dedicated section.

From an application perspective, the inherent optimality properties are more important than their derivation. We therefore preview that the Kalman filter provides the optimal solution of the Bayesian filtering recursions under the assumption of Gaussian initial state and noise signals. The estimate is furthermore the *mean*, the *mode*, and the *median* of the related filtering distribution, each of which minimize a different criterion of the estimation error, as discussed in Anderson and Moore (1979) and Maybeck (1979). Moreover, it is the *unbiased linear minimum variance estimator* for linear systems with arbitrary noise as long as the noise mean and covariance values are available, as pointed out in Anderson and Moore (1979). This property can be found in various notions in literature and implies that any filter that provides a lower variance (if there is one) must be a nonlinear filter.

The considered system

For convenience, we restate the system equations (2.3):

$$x_{k+1} = F_k x_k + v_k, \tag{3.1a}$$

$$y_k = H_k x_k + e_k. \tag{3.1b}$$

The noise signals and the initial state have known statistics that can be compactly written as

$$\mathrm{E}\left\{\begin{bmatrix} x_0 \\ v_k \\ e_k \end{bmatrix}\right\} = \begin{bmatrix} \hat{x}_0 \\ 0 \\ 0 \end{bmatrix}, \quad \operatorname{cov}\left\{\begin{bmatrix} x_0 \\ v_k \\ e_k \end{bmatrix}, \begin{bmatrix} x_0 \\ v_l \\ e_l \end{bmatrix}\right\} = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & \delta_{kl} Q_k & 0 \\ 0 & 0 & \delta_{kl} R_k \end{bmatrix}. \tag{3.1c}$$

The Kronecker delta δ_{kl} is only nonzero for $k = l$ and is used to illustrate that process and measurement noise are white sequences. R_k is furthermore assumed to be positive definite, which implies that all components of the measurement y_k are uncertain. In a more general form, there can be a correlation between v_k and e_k. We, however, omit this case and refer the reader to Kailath et al. (2000) for details.

The actual algorithm

The Kalman filter equations can be stated in many different forms of which we present a scheme with alternating time and measurement update. At time k, the filtered state is represented by an estimate $\hat{x}_{k|k}$ and its covariance matrix $P_{k|k}$.

Suppose we have available $\hat{x}_{k|k}$ and $P_{k|k}$ which were obtained by processing all measurements $y_{1:k}$. Then, the *time update* or *prediction* gives an estimate of the state at $k+1$. The predicted estimate $\hat{x}_{k+1|k}$ and covariance $P_{k+1|k}$ are given by

$$\hat{x}_{k+1|k} = F_k \hat{x}_{k|k}, \tag{3.2a}$$

$$P_{k+1|k} = F_k P_{k|k} F_k^{\mathrm{T}} + Q_k. \tag{3.2b}$$

The calculation of $\hat{x}_{k+1|k}$ can be interpreted as performing the expectation of (3.1a) over the random variables x_k and v_k. $P_{k+1|k}$ represents the related covariance which is, again, an expected value.

The *measurement update* or *correction* involves the following computations based on previously predicted $\hat{x}_{k|k-1}$ and $P_{k|k-1}$:

$$S_k = H_k P_{k|k-1} H_k^{\mathrm{T}} + R_k, \tag{3.3a}$$

$$K_k = P_{k|k-1} H_k^{\mathrm{T}} S_k^{-1}, \tag{3.3b}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - H_k \hat{x}_{k|k-1}), \tag{3.3c}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}. \tag{3.3d}$$

The filtered state estimate $\hat{x}_{k|k}$ is a combination of $\hat{x}_{k|k-1}$ and the scaled *residual* $y_k - H\hat{x}_{k|k-1}$, which is the discrepancy between the observed output and its prediction. The matrix S_k is called *residual covariance* and can be interpreted as uncertainty of the predicted output. The correction factor K_k is commonly known as the *Kalman gain*. The covariance update (3.3d) can be re-formulated in several different ways, some of which are better in terms of preserving symmetry and positive definiteness. One specific update formula, the *Joseph form*, expresses $P_{k|k}$ as sum of two positive definite symmetric matrices, but also requires more matrix manipulations than (3.3d).

The algorithm is initialized by letting $\hat{x}_{0|0} = x_0$ and $P_{0|0} = P_0$, and proceeding with a time update.

Bayes optimal filter for the linear Gaussian case

Without digging much into the details, we here sketch a "first principles" derivation of the Kalman filter under the Gaussian assumption.

Given that the initial state x_0 and the process and measurement noise v_k and e_k are all Gaussian, it can be shown that the filtering and prediction densities are Gaussian as well. The results can be established by completing Example 2.1 with the details of the Gaussian distribution. We here highlight two intermediate steps that can be directly related to the Kalman filter recursions. Given a posterior density $p(x_k|y_{1:k}) = \mathcal{N}(x_k; \hat{x}_{k|k}, P_{k|k})$, the subsequent prediction density can be written as

$$p(x_{k+1}|y_{1:k}) = \mathcal{N}(x_{k+1}; F_k\hat{x}_{k|k}, F_kP_{k|k}F_k^{\mathrm{T}} + Q_k). \tag{3.4}$$

Mean and covariance of the prediction density relate directly to the Kalman filter time update (3.2).

Having established the above prediction, it is not difficult to derive the joint density of the predicted state and measurement. In terms of the predicted mean and covariance, $\hat{x}_{k|k-1}$ and $P_{k|k-1}$, we obtain

$$p(x_k, y_k|y_{1:k-1}) = \mathcal{N}\left(\begin{bmatrix} x_k \\ y_k \end{bmatrix}; \begin{bmatrix} \hat{x}_{k|k-1} \\ H_k\hat{x}_{k|k-1} \end{bmatrix}, \begin{bmatrix} P_{k|k-1} & P_{k|k-1}H_k^{\mathrm{T}} \\ H_kP_{k|k-1} & H_kP_{k|k-1}H_k^{\mathrm{T}} + R_k \end{bmatrix}\right). \tag{3.5}$$

The filtering density $p(x_k|y_{1:k-1}, y_k)$ can then be obtained by a conditioning step. The rules for the Gaussian distribution, summarized in Appendix A.2.1, can be applied and yield the Kalman filter measurement update of (3.3).

It should be noted that this simple derivation fails to establish the minimum variance property for arbitrary noise distributions with known mean and covariance. In fact, Kalman's original derivation did not rely on a Gaussian assumption but was based on the concept of *orthogonal projections* instead.

Intuitive interpretation

For readers who have not previously encountered the Kalman filter, it is instructive to investigate the simple mechanisms that compose the recursions (3.2) and (3.3). In the time update, the state is merely predicted using the dynamic model. The omitted process noise in (3.2a) is accounted for by increasing the state estimate covariance by Q_k in (3.2b). In the measurement update, the state is corrected using the residual that is scaled by K_k. An increase in R_k gives a decrease in S_k and consequently K_k, which yields an $\hat{x}_{k|k}$ that relies heavier on the prediction $\hat{x}_{k|k-1}$. It is obvious from the covariance correction (3.3d), that (with an abuse of notation) $P_{k|k} \leq P_{k|k-1}$. Informally speaking, the prediction blows up $P_{k|k}$ to obtain $P_{k+1|k}$, while the correction reduces it again to give $P_{k+1|k+1}$, and so forth.

For further intuitive insights, the reader is referred to Chapter 1 of Maybeck (1979), where a simple scalar example is presented in a step-by-step fashion.

Further remarks

In many sources, an optimal one-step-ahead predictor rather than filter is derived, for instance Anderson and Moore (1979). This is justified by simpler equations from a least-squares perspective. Moreover, the alternating measurement and time update equations can be condensed into one equation each for $\hat{x}_{k+1|k}$ and $P_{k+1|k}$. Substituting K_k and S_k in (3.3d) yields

$$P_{k|k} = P_{k|k-1} - P_{k|k-1}H_k^{\mathrm{T}}(H_kP_{k|k-1}H_k^{\mathrm{T}} + R_k)^{-1}H_kP_{k|k-1}. \tag{3.6}$$

Plugging the above $P_{k|k}$ into (3.2b) gives a condensed update

$$P_{k+1|k} = F_kP_{k|k-1}F_k^{\mathrm{T}} - F_k^{\mathrm{T}}P_{k|k-1}H_k^{\mathrm{T}}(H_kP_{k|k-1}H_k^{\mathrm{T}} + R_k)^{-1}H_kP_{k|k-1}F_k^{\mathrm{T}} + Q_k. \tag{3.7}$$

The matrix relation (3.7) is of a specific form, namely a *discrete-time Riccati equation*, and needs to be solved forwards in time in the filtering context. Such equations establish a duality between optimal estimation and optimal control, where the latter requires solving a Riccati equation backwards in time. Kalman addressed these relations in his seminal paper, Kalman (1960).

The Kalman filter provides the means to treat non-stationary signals for the filtering problem by working in time domain, and in that respect extends the Wiener filter. Someone might remark, though, that some signals with given spectra cannot be represented in state space form.

Another interesting fact is that all presented matrix updates (3.2b) and (3.3d) do not rely on y_k. As a consequence, $P_{k|k}$ can be computed offline for any k and does not depend on the specific realization of the measurements.

3.1.3 Further Developments

The development of the Kalman filter was quickly followed by many extensions. The following paragraphs cover some of these extensions that apply to estimation problems in linear systems.

The time-invariant Kalman filter

For a linear time-invariant system with $F_k = F$, $H_k = H$, $Q_k = Q$, and $R_k = R$, it can be easily shown that the matrices $P_{k|k}$ and $P_{k+1|k}$ converge to constant values. This implies that the Kalman gain approaches a constant value. The Riccati equation (3.7), can be re-written using $P_{k+1|k} = P_{k|k-1} = P$ to yield

$$P = FPF^{\mathrm{T}} - F^{\mathrm{T}}PH^{\mathrm{T}}(HPH^{\mathrm{T}} + R)^{-1}HPF^{\mathrm{T}} + Q. \tag{3.8}$$

The matrix relation (3.8) is a *discrete-time algebraic Riccati equation* and can be solved for P. The resulting stationary Kalman gain K is given by

$$K = PH^{\mathrm{T}}(HPH^{\mathrm{T}} + R)^{-1} \tag{3.9}$$

and replaces the time-variant K_k in (3.3c). Much of Kalman's original work was on the stationary filter, which is a powerful tool if computational resources prohibit the use of a time-variant filter. Anderson and Moore (1979) dedicates an entire chapter to such time-invariant filters. The mathematically inclined reader is furthermore referred to Kailath et al. (2000) for a review of Riccati equations.

The information filter

We provided the filter recursions (3.2) and (3.3) in terms of a state estimate and its covariance matrix. An equivalent algorithm can be developed, that works with an inverse of the covariance matrix and a linearly transformed state estimate. Such algorithms are known as Kalman filters in information form, or *information filters*. Their use turns out to be computationally beneficial when many measurements are present. Further details can be found in Anderson and Moore (1979).

Square root implementations

The Kalman filter recursions involve multiplication and inversion of matrices in order to update covariance matrices. Due to numerical issues in these sensitive matrix operations, some of the covariance matrices might lose symmetry or positive definiteness. These problems can be addressed by working with matrix square roots of the covariance matrices instead. The resulting algorithms are known as *square root implementations* of the Kalman filter, and make up a considerable portion of the standard literature. The reader is referred to Anderson and Moore (1979) or Kailath et al. (2000) for details, where the latter reference uses the term *array algorithms*.

Prediction and smoothing

In Section 2.2, we introduced the different estimation problems filtering, prediction, and smoothing. In the Kalman filter framework we encountered already one-step-ahead prediction, which is often used to derive the algorithm. A simple way to predict more than one time instance is to repeatedly carry out the time update (3.2). Smoothing is a bit more complicated and there exist several approaches. The fixed-interval smoothing algorithm that is often referred to as Kalman smoother was developed by Rauch et al. (1965). Their suggested algorithm amounts to a Kalman filter that is run forward in time, and an extra backwards pass. An extensive treatment of several smoothing algorithms can be found in Kailath et al. (2000).

3.2 Kalman Filters for Nonlinear Systems

We here present a number of algorithms which have the Kalman filter as common ancestor. They all employ a measurement update similar to (3.3c), that is: a relation which is linear in the residual. The core of each algorithm is the way in which certain expected values, such as the mean and covariance of the predicted state, are computed.

Although we here list the methods that each specific Kalman filter uses, a thorough discussion is postponed to Chapter 4. The reason is that the extensive treatment of expected values there goes beyond the use in Kalman filtering algorithms.

Sections 3.2.2 through 3.2.4 list some of the most common Kalman filter variants and outline the techniques that are utilized to compute the required expected values. Directions to the corresponding sections in Chapter 4 are given for each algorithm. We roughly divide the approaches into *extended Kalman filters* and those that are based on *numerical integration*. This is one specific way of categorizing, but others are also conceivable. For instance, van der Merwe (2004) groups algorithms into those that are based on sampling and others. As we will see, there are always relations and connections that can be drawn. In the end, however, all of the listed algorithms have their origin in the Kalman filter.

3.2.1 A Common Underlying Concept

We presented an easy way to derive the Kalman filter as optimal Bayes estimator for a linear Gaussian system in Section 3.1.2. This “first principles” derivation relies on the assumption that the predicted state and the predicted measurement are jointly Gaussian distributed and obtains the equations for the measurement update (3.3) from the rules for conditional Gaussian distributions. For nonlinear systems, even if the noise sources are Gaussian, the joint prediction density $p(x_k, y_k | y_{1:k-1})$ is certainly not Gaussian.

However, given that we have the following Gaussian approximation:

$$p(x_k, y_k | y_{1:k-1}) \approx \mathcal{N}\left(\begin{bmatrix} x_k \\ y_k \end{bmatrix}; \begin{bmatrix} \hat{x}_{k|k-1} \\ \hat{y}_{k|k-1} \end{bmatrix}, \begin{bmatrix} P_{k|k-1} & M_k \\ M_k^{\mathrm{T}} & S_k \end{bmatrix} \right), \tag{3.10}$$

we can perform the measurement update of the Kalman filter (3.3). Such approaches also go by the name of *assumed density filtering*, as in Maybeck (1979).

The essence of the above paragraph is the following: if we can compute the parameters of the joint density (3.10), then we can apply a Kalman filter. The quantities $x_{k|k-1}$, $y_{k|k-1}$, $P_{k|k-1}$, M_k, and S_k are all expected values, and their computation is at the very heart of any nonlinear Kalman filter. We will now specify the involved moment integrals for time and measurement update.

Time update

The mean and covariance computations of the predicted state state amount to calculating the following expected values in the general (possibly nonlinear) case:

$$\begin{aligned}
\hat{x}_{k+1|k} &= \mathrm{E}\left\{x_{k+1}|y_{1:k}\right\} \\
&= \iint f_k(x_k, v_k) p(x_k, v_k|y_{1:k}) \, \mathrm{d}x_k \, \mathrm{d}v_k, && \text{(3.11a)} \\
P_{k+1|k} &= \operatorname{cov}\left\{x_{k+1}|y_{1:k}\right\} \\
&= \iint \Big(f_k(x_k, v_k) - \hat{x}_{k+1|k}\Big)\Big(\,\cdot\,\Big)^{\mathrm{T}} p(x_k, v_k|y_{1:k}) \, \mathrm{d}x_k \, \mathrm{d}v_k. && \text{(3.11b)}
\end{aligned}$$

Here, we introduced the compact notation $(z)(\,\cdot\,)^{\mathrm{T}}$ to abbreviate outer products $(z)(z)^{\mathrm{T}}$. The reader can easily verify that these integrals amount to the Kalman filter time update (3.2) for the linear case.

Measurement update

The predicted output and its covariance require integration with respect to the measurement noise e_k. The expressions are given by

$$\begin{aligned}
\hat{y}_{k|k-1} &= \mathrm{E}\left\{y_k|y_{1:k-1}\right\} \\
&= \iiint h_k(f_k(x_{k-1}, v_{k-1}), e_k) p(x_{k-1}, v_{k-1}, e_k|y_{1:k-1}) \, \mathrm{d}x_{k-1} \, \mathrm{d}v_{k-1} \, \mathrm{d}e_k, \\
& && \text{(3.12a)} \\
S_k &= \operatorname{cov}\left\{y_k|y_{1:k-1}\right\} \\
&= \iiint \Big(h_k(f_k(x_{k-1}, v_{k-1}), e_k) - \hat{y}_{k|k-1}\Big)\Big(\,\cdot\,\Big)^{\mathrm{T}} \\
&\qquad p(x_{k-1}, v_{k-1}, e_k|y_{1:k-1}) \, \mathrm{d}x_{k-1} \, \mathrm{d}v_{k-1} \, \mathrm{d}e_k. && \text{(3.12b)}
\end{aligned}$$

The cross covariance matrix between predicted state and measurement is given by the integral

$$\begin{aligned}
M_k &= \operatorname{cov}\left\{x_k, y_k|y_{1:k-1}\right\} \\
&= \iiint \Big(f_k(x_k, v_k) - \hat{x}_{k|k-1}\Big)\Big(h_k(f_k(x_{k-1}, v_{k-1}), e_k) - \hat{y}_{k|k-1}\Big)^{\mathrm{T}} \\
&\qquad p(x_{k-1}, v_{k-1}, e_k|y_{1:k-1}) \, \mathrm{d}x_{k-1} \, \mathrm{d}v_{k-1} \, \mathrm{d}e_k. && \text{(3.12c)}
\end{aligned}$$

All of the above expected values in (3.12) are with respect to a previously obtained posterior of the state, that is the density of $x_{k-1}|y_{k-1}$. It is, however, common in literature to first compute (3.11) and to use $\hat{x}_{k|k-1}$ and $P_{k|k-1}$ to define an intermediate prediction density $p(x_k|y_{1:k-1})$, or rather $p(x_k, e_k|y_{1:k-1})$ in general. The subsequent calculations of $\hat{y}_{k|k-1}$, S_k, and M_k are simplified in the sense that the process noise v_{k-1} vanishes from the equations. Furthermore, the integration is with respect to x_k rather than x_{k-1}:

$$\hat{y}_{k|k-1} = \iint h_k(x_k, e_k) p(x_k, e_k | y_{1:k-1}) \,\mathrm{d}x_k \,\mathrm{d}e_k, \tag{3.13a}$$

$$S_k = \iint \Big(h_k(x_k, e_k) - \hat{y}_{k|k-1}\Big) \Big(\cdot\Big)^{\mathrm{T}} p(x_k, e_k | y_{1:k-1}) \,\mathrm{d}x_k \,\mathrm{d}e_k, \tag{3.13b}$$

$$M_k = \iint \Big(x_k - \hat{x}_{k|k-1}\Big) \Big(h_k(x_k, e_k) - \hat{y}_{k|k-1}\Big)^{\mathrm{T}} p(x_k, e_k | y_{1:k-1}) \,\mathrm{d}x_k \,\mathrm{d}e_k. \tag{3.13c}$$

For the linear Gaussian case of Section 3.1, we can identify $M_k = P_{k|k-1} H_k^{\mathrm{T}}$ and $S_k = H_k P_{k|k-1} H_k^{\mathrm{T}} + R_k$.

Even if the considered systems are nonlinear, the measurement update of any Kalman filter remains a linear correction, that is: linear in the residual, of the prediction $\hat{x}_{k|k-1}$. Having computed the above statistics, the measurement update is given by

$$K_k = M_k S_k^{-1}, \tag{3.14a}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}), \tag{3.14b}$$

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^{\mathrm{T}}. \tag{3.14c}$$

Here, (3.14c) is another popular formulation of the posterior covariance matrix. It can be easily verified that the Kalman filter correction (3.3) is obtained for a linear system.

3.2.2 Extended Kalman Filters

The following paragraphs summarize a number of algorithms that can be related to the *extended Kalman filter*, or EKF. The central idea is to simplify the system equations in order to obtain problems that admit a simpler solution. Covered are the "standard" EKF, variants that go beyond linearization of the dynamic model, and variants that are based on divided differences and have an interpolation interpretation. The corresponding moment computation schemes are detailed in Section 4.5.

Linearized and extended Kalman filter

The Kalman filter can be easily applied to any linear system with knowledge about the involved noise signals. Therefore, a reasonable idea is to apply it to linearized version of a nonlinear system.

For applications in which computational resources are scarce but a nominal state trajectory $\bar{x}_{1:k}$ is available, the system can be linearized offline about each $\bar{x}_k$. The result is a linear time-varying system, which is then used in the *linearized Kalman filter*. If the nominal trajectory deviates from the true $x_{1:k}$ that we would like to estimate, the linearization might not reflect the true system well enough for the filter to perform well.

The EKF is an extension of the above idea that had its dawn not long after the seminal paper Kalman (1960). NASA scientist Schmidt and his colleagues applied the Kalman filter to a consecutively linearized model, where the linearization points are state estimates. The linearization is carried out about a filtered estimate $\hat{x}_{k|k}$ for the time update, and about a prediction $\hat{x}_{k|k-1}$ for the measurement update. Interestingly, this nonlinear extension and application in the Apollo program is what made the "linear" Kalman filter of Section 3.1 so popular.

For a nonlinear system with zero-mean additive noise (2.2), the EKF time update has the following form

$$\hat{x}_{k+1|k} = f_k(\hat{x}_{k|k}), \tag{3.15a}$$

$$P_{k+1|k} = F_k(\hat{x}_{k|k})P_{k|k}(F_k(\hat{x}_{k|k}))^{\mathrm{T}} + Q_k. \tag{3.15b}$$

The matrix $F_k(\hat{x}_{k|k})$ denotes the Jacobian matrix of $f_k(x_k)$, that contains first order partial derivatives, evaluated at $\hat{x}_{k|k}$. The relation to the "standard" Kalman filter prediction (3.2) are apparent. The EKF measurement update is equally close to the Kalman filter correction (3.3), and therefore here omitted. The only difference to (3.3) is that the predicted output $H_k\hat{x}_{k|k-1}$ is replaced by $h_k(\hat{x}_{k|k-1})$. Furthermore, the matrix H_k is in all matrix equations replaced by another Jacobian matrix $H_k(\hat{x}_{k|k-1})$, that contains partial derivatives of $h_k(x_k)$ evaluated at $\hat{x}_{k|k-1}$.

An iterative scheme to improve estimation results is based on the fact that the linearization in the measurement update is about $\hat{x}_{k|k-1}$, a predicted state. The main idea is that a more exact linearization of $h_k(x_k)$ should be obtained by evaluating its Jacobian at $\hat{x}_{k|k}$. Thus, the entire EKF measurement update is carried out once more, this time with $H_k(\hat{x}_{k|k})$. This scheme can be executed repeatedly and is known as *iterated* EKF.

For a more extensive description of the EKF equations, including a treatment of general models (2.1), the reader is referred to Hendeby (2008).

Higher order EKF

Going beyond mere linearization, it is reasonable to ask whether the system can be approximated better by Taylor polynomials of higher degree. Such system approximations require calculation of higher order partial derivatives, hence the title of this section.

Especially filters that are based on quadratic approximations to all nonlinearities in the state space equations (2.1) have a long history. Early references are Athans et al. (1968) and Sorenson and Stubberud (1968), with the former working on continuous-time dynamics. We refer to the resulting algorithm *second order* EKF, or simply EKF2. The term EKF with *bias compensation* is also common, as in Bar-Shalom et al. (2001). For a general system (2.1), the algorithm is listed in Hendeby (2008) or Roth and Gustafsson (2011).

One of the shortcomings of the EKF2 is the vastly increased computational complexity when implemented without extra care. However, there are ways to alleviate such shortcomings to some extend. Also, there exists a computationally cheaper EKF2 version that is based on function evaluates, similar to the numerical integration filters of the next section. The details are given in a paper by the present author: Roth and Gustafsson (2011) and further explained in Section 4.6. Furthermore, we there show some interesting parallels to the *unscented transformation*.

Function approximations beyond Taylor polynomials of degree two are in principle possible. However, such high order approximations are rarely seen in practice due to the seemingly complicated form the required expected values (3.11–3.13) assume. Section 4.5 presents a convenient reformulation of the Taylor series that simplifies the moment calculations and hence facilitates high order Kalman filters that are less dreadful. Other high order EKF variants are treated in Luca et al. (2006), for scalar systems, and Baum et al. (2011), where a specific formula is used to compute higher from lower moments.

Divided difference EKF

The Taylor series expansion of functions in nonlinear state space models requires differentiability. Furthermore, the functions must be available as analytical expressions in the first place. This is not always the case and considered as one potential drawback of the above EKF variants. An idea to alleviate this problem is to use *divided difference* approximations to the derivatives in the Taylor series or linearization. As soon as the involved functions are available for evaluation at specific points, these can be calculated. Another interpretation of such a differencing scheme is *interpolation* of the involved nonlinearities.

A linearization method that is based on divided differences and its use in Kalman filtering was first suggested in Schei (1997). One inherent feature is that the uncertainty of the state and the noise can be taken into account when finding perturbation points for the differencing. A second order finite difference scheme was suggested separately by Ito and Xiong (2000) and Nørgaard et al. (2000b,a). In both papers, a number of terms are sacrificed for the sake of a faster algorithm. Also, the efficient EKF2 implementation of Roth and Gustafsson (2011) can be related to such second order divided difference filters.

Because the divided difference EKF variants are based on processed function evaluates, there is an apparent relation to the filters of the next section. In fact, the term *sigma point Kalman filters* is popular for all sampling based approaches. The thesis by van der Merwe (2004) is focused on sigma point filters, but also more recent publications emphasize the links: Gustafsson and Hendeby (2012) and Šimandl and Duník (2009).

3.2.3 The Unscented and Numerical Integration Kalman Filters

This section introduces the *unscented Kalman filter* or UKF, and a number of related algorithms that are based on numerical integration of the expected value integrals (3.11–3.13). While the UKF is based on an appealing intuitive idea, the latter has a solid theoretical foundation including *Gauss-Hermite quadrature*. Both UKF and the numerical integration filters are based on deterministic sampling and sometimes subsumed with the divided difference filters of Section 3.2.2 as *sigma point Kalman filters*. The utilized moment computation schemes are presented in Section 4.4.

The unscented Kalman filter

The unscented Kalman filter is popular among some, others tend to avoid it. At least this is one interpretation of the fact that the first journal paper Julier et al. (2000) was not published until several years after initial submission, whereas the paper Julier and Uhlmann (2004) was an invited contribution. The thesis van der Merwe (2004) provides a summary of different UKF variants.

The filter uses the *unscented transformation* of Section 4.4 to compute the required expected values in (3.11–3.13). In doing so, merely function evaluates are required, which makes the filter simple to implement. Without the need to calculate derivatives, as in EKF, differentiability issues are avoided. The UKF has been shown to outperform the standard EKF in problems where linearization is just not sufficient, at roughly the same computational cost.

The underlying unscented transformation is sometimes criticized for its lack of theoretical justification. As result of a staged development throughout several papers, from Julier et al. (1995) to Julier (2002), the algorithm requires selecting parameters that are not very intuitive. Furthermore, the accuracy statements in Julier et al. (2000) are difficult to assess and easily misinterpreted. We address these issues in Section 4.4.

The UKF has found its way into different branches of robotics, Lefebvre et al. (2005) and Thrun et al. (2005), and has been also applied to provide the proposal density of a particle filter in van der Merwe (2004). For one specific parameter choice, the *cubature Kalman filter* of the next section is obtained.

Numerical integration Kalman filters

Next, we introduce filters that assume the nonlinear state space model (2.2), with independent Gaussian process and measurement noise. In this case, the expected values in (3.11–3.13) all involve integration of functions with exponential weight $\exp(-x^{\mathrm{T}}x)$ over the entire n-dimensional space. Approximately computing such weighted integrals has been studied by mathematicians for centuries, and indeed concepts such as Gauss-Hermite quadrature are applied in the *numerical integration Kalman filters*. Specific computation schemes are further highlighted in Section 4.4.

Two Kalman filter variants have gained specific attention in the signal processing community. Common to them is that accuracy statements about the moment computation schemes are available: the chosen integration rules work for polynomial nonlinearities of a certain degree.

Gauss-Hermite Kalman filters, which employ Gauss-Hermite quadrature, are described in Ito and Xiong (2000), although related ideas were mentioned earlier in Sorenson (1974). Similar to UKF, the algorithm is sampling based. However, the number of deterministic samples increases exponentially with the state dimension. Consequently, the filters entail computational demands that render an application in high-dimensional problems unfeasible. An idea to alleviate this issue is given in Closas et al. (2012), where the authors suggest to break down the original filtering problem into smaller chunks. This requires a thorough analysis of the problem and might not always be possible.

The *cubature Kalman filter* or CKF of Arasaratnam and Haykin (2009) scales better in terms of computations. The integrals in (3.11–3.13) are here solved using an integration rule of degree 3, which means that the mean computations are exact for polynomial nonlinearities up to degree 3. The covariance matrices, however, are only guaranteed to be correct for linear systems. Further details are given in Section 4.4. The computational demand of CKF is the same as UKF, and actually CKF can be obtained by setting the UKF parameters in a certain way. Albeit UKF and CKF are very close to one another, CKF is preferred by some due to its sound theoretical derivation.

The numerical integration viewpoint has also been used to analyze the UKF in Wu et al. (2006). Beyond the fairly general idea to employ Gauss-Hermite quadrature in Ito and Xiong (2000), and the degree-3 rule of Arasaratnam and Haykin (2009), any other integration rule can be applied. Rules with degree 5 are explored in Jia et al. (2013), for instance.

3.2.4 Further Variants

Apart from the algorithms of the previous sections, there certainly exist many more Kalman filter variants. An attempt to compile a "complete" list is bound to fail.

Having understood the central dependence on the expected values (3.11–3.13), any method to solve these yields a specific Kalman filter algorithm. With sufficient computational resources, for instance, a Kalman filter based on *Monte Carlo integration* of (3.11–3.13) is conceivable. Some of the moment integrals may even admit a closed form solution. Different methods can furthermore be combined in order to optimize for accuracy and reduced computational demand, as suggested in Gustafsson and Hendeby (2012).

One approach that does not quite fit what has been listed before is the *Fourier-Hermite Kalman filter* of Sarmavuori and Särkkä (2012), which utilizes a *Fourier-Hermite series* to approximate the involved nonlinearities, as opposed to the Taylor series of EKF variants. The filter reduces to the *statistically linearized* filter of Gelb (1974) as special case. The formulas require a number of expected values to be calculated exactly, which stands out from the other approaches we present.

3.3 Particle Filters

Particle filters, or more general *sequential Monte Carlo methods*, are a conceptually different family of state estimation algorithms. In contrast to the mean and covariance computations of a Kalman filter, the entire filtering density $p(x_k|y_{1:k})$ is approximated throughout time. Furthermore, the algorithms cope with arbitrary system nonlinearities and arbitrary noise distributions. These advantages come at the price of a vastly increased computational complexity that prohibits the use of particle filters for larger state dimensions.

3.3.1 Background

The idea to use Monte Carlo methods for state estimation dates back to as early as Handschin and Mayne (1969). However, it was not until the seminal paper Gordon et al. (1993) that a fully working state estimation algorithm was introduced, that is now widely known as the particle filter.

Except for the research monograph Ristic et al. (2004) and the dedicated chapters in Gustafsson (2010b), there are few books that cover the particle filter. However, there exist many tutorial articles that provide an introduction to the topic: Arulampalam et al. (2002) has a convincing number of citations, Gustafsson (2010a) includes a selection of actual applications beyond academic examples, Cappé et al. (2007) goes beyond filtering and surveys several smoothing techniques as well.

The approximation of the entire filtering density is achieved by propagating a large number of random samples and weights throughout time. In the basic particle filter of Gordon et al. (1993), the way in which each of the samples or *particles* evolves is governed by the underlying dynamic model. The weights are computed based on the measurement model. At certain times this particle ensemble is rejuvenated by a, for the algorithm crucial, resampling step.

The simulation based particle filters can be applied to arbitrary state space models. The state and measurement spaces are not even restricted to be continuous but could be discrete sets. Also, the noise signals can admit arbitrary distributions as long as the model can be simulated. Multimodality, for instance, is not an issue. The limiting factor is, however, that only low-dimensional problems can be treated without further thought. The computational demand is in general higher than for a simple EKF and grows linearly with the number of particles.

3.3.2 Basic Algorithm

The particle filter applies to the general state space model (2.1), that we here restate for convenience:

$$x_{k+1} = f_k(x_k, v_k), \tag{3.16a}$$

$$y_k = h_k(x_k, e_k). \tag{3.16b}$$

The noise v_k and e_k, and the initial state x_0, can admit arbitrary distributions but are assumed independent. Furthermore, we assume that the likelihood $p(y_k|x_k)$ can be evaluated. Essentially, this requires that the measurement noise must be fully specified for any given x_k and y_k.

Particle approximation to the posterior

The filtering posterior is approximated by a sum of weighted Dirac impulses at N particles $x_k^{(i)}$:

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^N w_k^{(i)} \delta(x_k - x_k^{(i)}). \tag{3.17}$$

In order for (3.17) to be a valid probability density, the weights add up to one,

$$\sum_{i=1}^N w_k^{(i)} = 1. \tag{3.18}$$

The weights $w_k^{(i)}$ and particles $x_k^{(i)}$ are generated based on the principle of *importance sampling*, which implies that (3.17) approximates the true posterior when it comes to computing expected values. For example, the conditional mean, or MMSE estimate, is given by

$$\mathrm{E}\left\{x_k|y_{1:k}\right\} \approx \sum_{i=1}^N w_k^{(i)} x_k^{(i)}. \tag{3.19}$$

Actually, the particle filter approximates the joint density of the state trajectory,

$$p(x_{0:k}|y_{1:k}) \approx \sum_{i=1}^N w_k^{(i)} \delta(x_{0:k} - x_{0:k}^{(i)}), \tag{3.20}$$

which is computationally cheaper than working on the last state only. The details are described in Gustafsson (2010a) or Arulampalam et al. (2002). This particle approximation is usually only good for the most current states, including x_k. In the simplest setting, the marginal posterior (3.17) is obtained by discarding all past states $x_{0:k-1}$.

Filter recursions

At each time step k, the algorithms requires N particles $x_k^{(i)}$ to be drawn from *proposal distributions*,

$$x_k^{(i)} \sim q(x_k|x_{0:k-1}^{(i)}, y_{1:k}), \tag{3.21}$$

that might in principle depend on the entire particle history $x_{0:k-1}^{(i)}$ and all measurements $y_{1:k}$. In fact, the proposal might be obtained by a nonlinear Kalman filter that is run in parallel.

The weights are updated according to

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(x_k^{(i)}|x_{k-1}^{(i)}) p(y_k|x_k^{(i)})}{q(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})}. \tag{3.22}$$

and subsequently normalized. Equation (3.22) ensures that the approximation in (3.20) is meaningful. The reader is referred to Arulampalam et al. (2002) for a compact derivation.

Except for (3.21) and (3.22), the algorithm requires a resampling procedure that is described in a dedicated paragraph below.

The filter is initialized by drawing N samples $x_0^{(i)}$ from $p(x_0)$, and setting all weights to $1/N$.

The bootstrap filter

If the proposal density is chosen as $p(x_k|x_{k-1}^{(i)})$, the weight update simplifies significantly,

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(y_k|x_k^{(i)}), \tag{3.23}$$

because the weight update involves only the likelihood. Generating samples from $p(x_k|x_{k-1}^{(i)})$ amounts to simulating the state transition for an independent process noise realization,

$$x_k^{(i)} = f(x_{k-1}^{(i)}, v_{k-1}^{(i)}). \tag{3.24}$$

This yields the filter of Gordon et al. (1993), which is easily implemented in few lines of MATLAB code.

Resampling

The bare recursions in (3.21) and (3.22) yield no functioning estimation algorithm. The problem is that all but one weights are zero after few time steps, and referred to as *weight degeneracy*. In order to overcome this issue, a resampling step was introduced in Gordon et al. (1993): particles are discarded or duplicated according to their weights, either at each k or when required. The degree of degeneracy can be assessed by the *effective number of particles*, which is approximately given by

$$N_{\text{eff}} = \left(\sum_{i=1}^{N} (w_k^{(i)})^2\right)^{-1}. \tag{3.25}$$

Resampling can be applied whenever N_{eff} falls below a threshold.

A number of resampling algorithms are listed in Arulampalam et al. (2002).

Relation to other algorithms and further development

Gustafsson (2010a) highlights the relation between particle filters and *point mass filters* or PMF, an algorithm that dates back to the early days of nonlinear filtering and is surveyed in Sorenson (1974). The PMF computes the exact posterior for N spatially fixed grid points, which is not possible in high-dimensional problems due to the sheer number of required points. The particle filter, instead, uses an adaptive grid that is rejuvenated constantly. Furthermore, by "secretly" working on the joint smoothing density, the particle filter complexity is linear in the number of particles N, in contrast to the N^2 complexity of the PMF.

Particle filters cannot be applied to high-dimensional problems without further thought. The reason is that it becomes increasingly difficult to generate meaningful samples as the state dimension increases. So called *marginalized* or *Rao-Blackwellized* particle filters address this issue to some extend by running a particle filter only for those states of a system that are severely nonlinear. More information is, for instance, given in Gustafsson (2010b) or Hendeby et al. (2010).

A number of particle smoothing algorithms has emerged in recent years. The variants surveyed in Cappé et al. (2007) rely on the output of a previously run particle filter, in the same way the Rauch-Tung-Striebel smoother in Section 3.1 relies on a Kalman filter output. An overview of particle smoothing techniques is also given in Lindsten and Schön (2013).

3.3.3 Example Application: Road-Constrained Target Tracking

We here present an example that is difficult to approach using Kalman filter based methods, but that can be solved by straightforward application of the particle filter. The state space is *hybrid*: the state has discrete as well as continuous components. The continuous state components are subject to constraints, and the state transition and measurement functions in (3.16) require table look-up. Thus, the problem is nonlinear and far from Gaussian because of the integer state components.

The example is about improving GPS position data of road-bound vehicles by incorporating road network information. In automotive navigation systems, for example, it is useful to display the user position on a map. The GPS measurements, however, might be in conflict with the available map. On the one hand, this can be a result of an incorrect map. On the other hand, the GPS signals might be subject to multipath effects or atmospheric disturbances. We aim at generating road-bound positions from GPS data by processing them in a particle filter that utilizes *road coordinates*.

The material of this section is taken from Roth et al. (2012), where a particle filter and several particle smoothers are applied to GPS data that were collected in real world experiments.

Problem description

The GPS data are typically scattered about the roads, as illustrated in Figure 3.1. Considering the GPS measurements between 334 seconds and 380 seconds, it becomes obvious that generating on-road trajectories cannot be achieved by simply projecting each GPS datum onto the closest road segment. A filtering approach seems therefore reasonable.

Expressed as a filtering problem, we want to estimate the kinematic state of a target that is known to travel on-road by processing available GPS measurements. For simplicity, roads are abstracted as interconnected straight road segments of zero width. Specifically, each road segment is described by the indices of its starting and end points. These *nodes* of the road network are stored along their position and the nodes that they are connected to. The entity of all nodes fully describes the road network.

State space model in road coordinates

The aim of deriving the following model is its use in a particle filter algorithm.

Each on-road target can be described by a four-dimensional state vector. Two integers n_1 and n_2 identify the road segment, and form the discrete part of the state

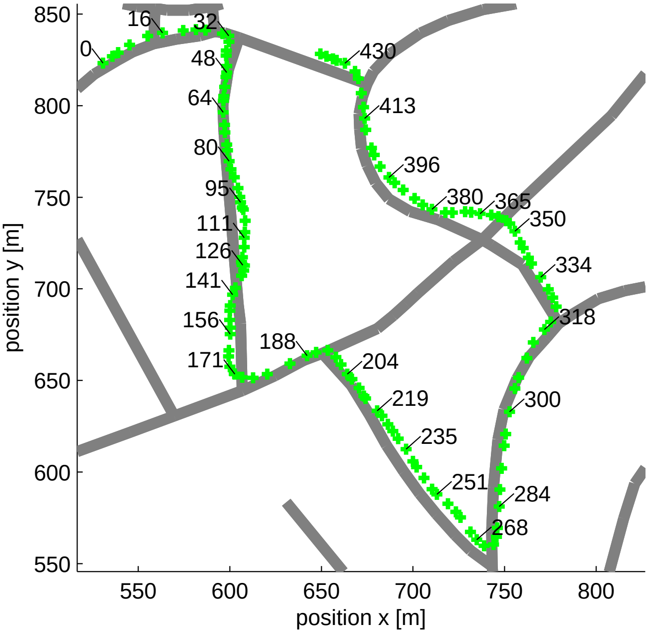


Figure 3.1: *GPS position measurements plotted on a section of the road network. The time stamps are in seconds.*

space. Given that the road segment is determined by n_1 and n_2, the target on that road segment is described by a distance d and a speed s. Here, some conventions must be met: the distance is measured from n_1, and a vehicle traveling from n_1 towards n_2 has positive speed. Obviously, d cannot exceed the length l of the current road segment. With l computed using n_1 and n_2 and a look-up of the node positions, we have the constraint $0 < \mathsf{d} < \mathsf{l}$.

As a consequence of the chosen states, the exact same state can be written in two different ways:

$$x_k = \left(\mathsf{d}_k, \quad \mathsf{s}_k, \quad \mathsf{n}_{1k}, \quad \mathsf{n}_{2k}\right)^T \triangleq \left(\mathsf{l}_k - \mathsf{d}_k, \quad -\mathsf{s}_k, \quad \mathsf{n}_{2k}, \quad \mathsf{n}_{1k}\right)^T. \tag{3.26}$$

Using the road network information, we can compute a *global position* $[\mathsf{x}_k, \mathsf{y}_k]^\mathrm{T}$ from each on-road state (3.26). With the GPS data in the same global coordinate frame, the measurement model (3.16b) follows immediately:

$$y_k = \left[\mathsf{x}_k^{\mathsf{GPS}}, \quad \mathsf{y}_k^{\mathsf{GPS}}\right]^T = h(x_k) + e_k. \tag{3.27}$$

The function $h(x_k)$ computes a global position from d_k, n_{1k} and n_{2k}. This requires a table look-up and is thus nonlinear. The two-dimensional measurement noise e_k remains to be specified.

The state transition function of (3.16a) involves first processing d_k and s_k, and subsequently adjusting the road segment if necessary. An update of the continuous components is given by a constant velocity model

$$\begin{pmatrix} \overline{\mathsf{d}}_{k+1} \\ \mathsf{s}_{k+1} \end{pmatrix} = \begin{pmatrix} 1 & \mathsf{T}_k \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathsf{d}_k \\ \mathsf{s}_k \end{pmatrix} + \mathsf{v}_k, \tag{3.28}$$

with $\mathsf{T}_k = \mathsf{t}_{k+1} - \mathsf{t}_k$ as sampling interval and t_k as time when the k-th GPS measurement is available. The process noise v_k is generated according to a distribution with covariance

$$\mathsf{Q}_k = \begin{pmatrix} \mathsf{T}_k^3/3 & \mathsf{T}_k^2/2 \\ \mathsf{T}_k^2/2 & \mathsf{T}_k \end{pmatrix} \mathsf{q}, \tag{3.29}$$

with q being a design parameter for tuning in the particle filter. The distribution of v_k is not restricted to be a Gaussian, but could be any distribution from which samples can be drawn. That includes the *t* distribution of Appendix A.3, which exhibits heavier tails than a Gaussian and can be used to model targets that maneuver heavily. More information on the utilized motion model is given in Bar-Shalom et al. (2001).

Now, simply applying (3.29) might yield a $\overline{\mathsf{d}}_{k+1}$ that exceeds the road segment: $\overline{\mathsf{d}}_{k+1} > \mathsf{l}_k$ or $\overline{\mathsf{d}}_{k+1} < 0$. If that is the case, the excess distance is projected onto one of the neighboring segments. When several adjacent road segments are encountered, one is picked randomly. This procedure involves table look-up and may require some iterations before an admissible d_{k+1} is found. The relation between x_{k+1} and x_k, that is (3.16a), is obviously highly nonlinear but can be easily simulated.

Particle filter in road coordinates

As explained in Section 3.3.2, the simplest particle filter uses $p(x_k|x_{k-1}^{(i)})$ as proposal density. This requires simulating the model for each particle, with an independent process noise realization. Thus, the dynamic model that we derived in the previous section lends itself to application in such a particle filter. The filter can be tuned by adjusting (3.29) so that simulated trajectories reflect realistic target behavior.

Next, we specify the likelihood of each particle $p(y_k|x_k^{(i)})$. With independent process noise in the measurement model (3.27), this amounts to specifying $p(e_k)$. From an inspection of Figure 3.1 it appears reasonable to have a likelihood that does not decay too fast with Euclidean distance (such that the filter works even though the measurement is far from the road), but also that does not assign disproportionately high weights to particles that are close to measurements. The latter demand is to avoid being tricked by ambiguous situations such as the period between between 350 and 380 seconds in Figure 3.1. The likelihood in Roth et al. (2012) is therefore chosen uniform on a circle around the origin, which gives the same result for all particles in a defined vicinity of the measurement. Particles that are too far from the measurement are assigned weight zero and automatically discarded in the next resampling step.

Having specified the dynamic model and the likelihood, all ingredients to run a particle filter are provided. Of course, a resampling scheme must be chosen. In Roth et al. (2012) it is shown that this yields indeed smooth on-road position trajectories. The utilized motion model is still very basic, but could be easily extended to account for vehicle stops, traffic lights, one-way roads, and speed limits. For further insights, actual results, and the treatment of smoothing algorithms for this example, the reader is referred to Roth et al. (2012).

4

Computing Expected Values of Transformed Random Variables

This chapter presents several techniques that can be used to approximately compute expected values of nonlinearly transformed random variables. There is an immediate relation to the nonlinear Kalman filters of Section 3.2, because the mean vectors and the covariance matrices, or *moments*, in (3.11–3.13) are such expected values. Thus, moment computations are the backbone of any nonlinear Kalman filter. Of course, the problem is also of relevance beyond a filtering context.

After stating the problem and giving some instructive examples, we specify a family of unimodal (not univariate) distributions that we mainly focus on: the elliptically contoured distributions of Section 4.2. Subsequently, a whole range of methods are discussed. That includes Monte Carlo methods in Section 4.3, integration rules based on deterministic sampling in Section 4.4, and methods that are based on simplifying the involved nonlinear functions in Section 4.5. Section 4.6 highlights connections among Section 4.4 and Section 4.5. Two illustrative examples are detailed in Section 4.7, and followed by some concluding remarks.

4.1 Motivation and Background

Our main motivation for investigating different computation methods for expected values is presented in the previous chapter, where we point out the importance of mean and covariance computations in nonlinear Kalman filter variants. The core equations are listed in (3.11–3.13). However, computing expected values or moments is a problem that attracts a broader audience than just signal processing researchers. An overview from the statistical perspective, for instance, is given in Evans and Swartz (1995).

Given infinite computational resources, the problems that we treat can in most cases be solved using Monte Carlo methods. Investigating alternatives, however, can be beneficial in several ways. Some deterministic methods are very efficient in terms of computations. Hence, they can be used in high-dimensional problems, and online applications where Monte Carlo methods might be too slow. Other methods can be used to gain insight about the involved nonlinearities.

The moment computation problem

The ingredients to the specific problem that we aim at solving are a known function $f : \mathbb{R}^n \to \mathbb{R}^{n_z}$ and the n-dimensional random vector x. We assume that x has a known probability distribution with density $p(x)$, or that x has at least known first moments

$$\mathrm{E}\{x\} = \hat{x}, \quad \mathrm{cov}\{x\} = P. \tag{4.1}$$

Given the above information, we would like to compute statistical information of the transformed random variable

$$z = f(x). \tag{4.2}$$

It might not be possible, or even desirable, to compute the density of the transformed variable $z = f(x)$. Instead, our focus is on the moments of z. These are nothing but expected values that convey relevant statistical information. Most important are the first two moments, or the mean value and the covariance matrix, that are given by the integrals

$$\hat{z} = \mathrm{E}\{f(x)\} = \int f(x) p(x) \,\mathrm{d}x, \tag{4.3a}$$

$$P_z = \mathrm{cov}\{f(x)\} = \int \Big(f(x) - \hat{z}\Big)\Big(f(x) - \hat{z}\Big)^{\mathrm{T}} p(x) \,\mathrm{d}x. \tag{4.3b}$$

Computing $\hat{z}$ and P_z in (4.3) is the central problem of this chapter and henceforth compactly referred to as the *moment computation problem*.

Actually, all components of $\hat{z}$ and P_z require solving scalar problems. For example, the l, m-th component of P_z requires computing

$$\mathrm{cov}\{f_l(x), f_m(x)\} = \int (f_l(x) - \hat{z}_l)(f_m(x) - \hat{z}_m) p(x) \,\mathrm{d}x, \tag{4.4}$$

where $f_l(x)$ and $f_m(x)$ denote row l and m of the vector valued function f. Of course, (4.4) still requires integration over the entire $\mathbb{R}^n$.

The reason to list the covariance matrix P_z separately is its frequent use in Gaussian approximations of $z = f(x)$. Especially in a Kalman filter context, it is common to assume $z \sim \mathcal{N}(\hat{z}, P_z)$, with $\hat{z}$ and P_z computed by some approximate moment computation scheme.

Beyond mean and covariance, it is possible to investigate higher moments of $z = f(x)$. Other than for a scalar z, this is very cumbersome. The sheer number of moments increases immensely with growing n_z: there are n_z first moments, n_z^2

second moments, and so on.

The covariance matrix (4.3b) can be re-written to obtain

$$P_z = \int f(x) f(x)^{\mathrm{T}} p(x)\,\mathrm{d}x - \hat{z}\hat{z}^{\mathrm{T}}. \tag{4.5}$$

Both (4.3b) and (4.5) rely on $\hat{z}$ that must be computed from (4.3a). If the utilized $\hat{z}$ is not exact in the first place, and if the utilized computation schemes are not exact for neither (4.3b) nor (4.5), then the obtained P_z can differ from the true value. The differences are difficult to assess without a hands-on example, but the reader should be aware of them. Moreover, the subtraction in (4.5) may yield a negative definite matrix P_z, why computation of (4.3b) might be preferable.

The methods that we discuss are mostly focused on "simple" unimodal densities $p(x)$. Nevertheless, the examined techniques can be extended to treat mixtures of unimodal distributions. Another reason to primarily work with unimodal distributions is the application in the approximate Kalman filters of Section 3.2. Although we sometimes focus on Gaussian random variables x, more general results for the family of elliptically contoured distributions are also discussed in Section 4.2.

Two instructive examples

We next introduce two instances of the moment computation problem (4.3). The first example relates to coordinate changes in state space models, as explained in Section 2.5. The second example is about approximating probability density functions.

Example 4.1: Kalman filter initialization

Suppose that we want to run a Kalman filter in a linear state space model with state x. In order to initialize the algorithm, we require the mean and covariance of the initial state x_0, that is $\hat{x}_0$ and P_0.

Now, suppose that the only available information is in a different coordinate frame z with $x = \tau(z)$, and suppose that we have available the density $p(z_0)$. Then, the sought statistics can be computed using

$$\hat{x}_0 = \mathrm{E}\left\{\tau(z_0)\right\} = \int \tau(z_0) p(z_0)\,\mathrm{d}z_0, \tag{4.6a}$$

$$P_0 = \operatorname{cov}\left\{\tau(z_0)\right\} = \int \big(\tau(z_0) - \hat{x}_0\big)\big(\tau(z_0) - \hat{x}_0\big)^{\mathrm{T}} p(z_0)\,\mathrm{d}z_0. \tag{4.6b}$$

Clearly, this is one instance of the problem in (4.3).

Example 4.2: Kullback-Leibler divergence

Sometimes it is required to approximate one probability density by another density that might be easier to handle in the specific application. In order to preserve as many of the original characteristics as possible, it is useful to employ *information theoretic criteria*. These facilitate comparing probability densities. The

Kullback-Leibler divergence, as presented in Bishop (2006), is popular in this context and given as functional of two densities $p(x)$ and $q(x)$:

$$\begin{aligned}\mathrm{KL}(p\|q) &= -\int p(x)\ln\left(\frac{q(x)}{p(x)}\right)\,\mathrm{d}x \\ &= -\int p(x)\ln\left(q(x)\right)\,\mathrm{d}x + \int p(x)\ln\left(p(x)\right)\,\mathrm{d}x. \end{aligned} \tag{4.7}$$

Obviously, the computation of $\mathrm{KL}(p\|q)$ involves computing expectations with respect to $p(x)$.

Now, suppose that $p(x)$ is given and that we want to minimize $\mathrm{KL}(p\|q)$ by choosing the parameters that determine $q(x)$. This amounts to minimizing the first term of (4.7), because the second term does not depend on $q(x)$. If $q(x)$ is determined by only one or two scalar parameters, a numerical search algorithm can be applied. However, it is required that the integral

$$\int f(x)p(x)\,\mathrm{d}x = \int \ln\left(q(x)\right)p(x)\,\mathrm{d}x \tag{4.8}$$

can be evaluated efficiently for any parameter setting. Here, the ability to efficiently compute an expected value of the form (4.3a) facilitates solving a minimization problem that might not be solvable otherwise.

The second example is continued for a specific choice of $p(x)$ and $q(x)$ in Section 4.7.2, and used as benchmark problem to compare a number of moment computation schemes.

4.2 Elliptically Contoured Distributions

We here generalize a number of results that are well known for the Gaussian case to a wider class of distributions, namely *elliptically contoured distributions*. The presentation follows the corresponding sections in Anderson (2003). The examples require knowledge of a number of probability distributions that are given in the Appendices A.2 and A.3.

The density of an n-dimensional Gaussian random vector with mean $\hat{x}$ and positive definite covariance P,

$$\mathcal{N}(x;\hat{x},P) = \frac{1}{(2\pi)^{\frac{n}{2}}}\frac{1}{\sqrt{\det(P)}}\exp\left(-\frac{1}{2}(x-\hat{x})^{\mathrm{T}}P^{-1}(x-\hat{x})\right), \tag{4.9}$$

is constant for elliptical regions described by

$$(x-\hat{x})^{\mathrm{T}}P^{-1}(x-\hat{x}) = c, \tag{4.10}$$

for some constant $c > 0$. In two dimensions, the density is constant on ellipses around the mean $\hat{x}$.

This result does not only hold for the Gaussian but also for any other density in which x solely enters in a quadratic form $(x-\hat{x})^{\mathrm{T}}P^{-1}(x-\hat{x})$. In general, such densities can be written as

$$p(x) = \frac{1}{\sqrt{\det(P)}} h\Big((x-\hat{x})^{\mathrm{T}} P^{-1} (x-\hat{x})\Big), \tag{4.11}$$

with a function $h : \mathbb{R} \to \mathbb{R}$ for which the n-dimensional integral

$$\int h(y^{\mathrm{T}} y)\,\mathrm{d}y = \int\!...\!\int h(y_1^2 + \ldots + y_n^2)\,\mathrm{d}y_1 ... \mathrm{d}y_n = 1. \tag{4.12}$$

The condition (4.12) is required for $p(x)$ to be a valid probability density function.

Example 4.3: Gaussian

For the n-dimensional Gaussian distribution, h is given by

$$h(z) = (2\pi)^{-\frac{n}{2}} \exp(-\frac{z}{2}), \tag{4.13}$$

and the factor $(2\pi)^{-\frac{n}{2}}$ ensures that the integral condition (4.12) is met. For the n-vector y, $h(y^{\mathrm{T}} y)$ is recognized as zero-mean Gaussian density with unity covariance.

Example 4.4: Student's t

For the n-dimensional t distribution with degrees of freedom ν, we have

$$h(z) = \frac{\Gamma(\frac{\nu+n}{2})}{\Gamma(\frac{\nu}{2})} \frac{1}{(\nu\pi)^{\frac{n}{2}}} \left(1 + \frac{1}{\nu} z\right)^{-\frac{n+\nu}{2}}. \tag{4.14}$$

Again, the relation to a "standard" t density with zero location and unity matrix parameter is apparent. $\Gamma(\,\cdot\,)$ denotes the Gamma function.

4.2.1 Stochastic Decoupling via a Coordinate Change

Suppose that we would like to compute expected values as in (4.3a), now with respect to the elliptically contoured density (4.11):

$$\mathrm{E}\{f(x)\} = \int f(x) \frac{1}{\sqrt{\det(P)}} h\Big((x-\hat{x})^{\mathrm{T}} P^{-1} (x-\hat{x})\Big)\,\mathrm{d}x. \tag{4.15}$$

We show that (4.15) can be re-written as an expected value of a different function with respect to a simpler density.

The key instrument in the following computations is a change of integration variables that involves a square root of the positive definite matrix P. Let U be such a *matrix square root* with

$$P = UU^{\mathrm{T}}, \tag{4.16}$$

for instance obtained by *Cholesky factorization*. Furthermore, consider the variable change relation

$$x = Uy + \hat{x}, \tag{4.17}$$

which implies $\mathrm{d}x = |\det(U)|\,\mathrm{d}y$, as known from substitution rules for multidimensional integrals. The quadratic form in (4.10) collapses after inserting (4.17):

$$(x-\hat{x})^{\mathrm{T}}P^{-1}(x-\hat{x}) = y^{\mathrm{T}}y. \tag{4.18}$$

The integral (4.15) simplifies to

$$\begin{aligned} \mathrm{E}\{f(x)\} &= \int f(Uy+\hat{x})\frac{1}{\det(U)}h(y^{\mathrm{T}}y)\det(U)\,\mathrm{d}y \\ &= \int f(Uy+\hat{x})h(y^{\mathrm{T}}y)\,\mathrm{d}y. \end{aligned} \tag{4.19}$$

Here, we used $\det(P) = \det(UU^{\mathrm{T}}) = \det(U)^2$. The integration region, being the entire $\mathbb{R}^n$, remains unchanged.

The implications for expected values of Gaussian and t vectors x are apparent and here illustrated for the Gaussian case. Any expected value of $f(x)$ with $x \sim \mathcal{N}(\hat{x}, P)$ can be re-written as

$$\mathrm{E}\{f(x)\} = \int f(x)\mathcal{N}(x;\hat{x},P)\,\mathrm{d}x = \int f(Uy+\hat{x})\mathcal{N}(y;0,I)\,\mathrm{d}y. \tag{4.20}$$

Sometimes, this technique is referred to as *stochastic decoupling* because the components of the random vector y are uncorrelated. It is a popular tool in many Kalman filter algorithms, from Schei (1997) to Arasaratnam and Haykin (2009).

4.2.2 A Change from Cartesian to Spherical Coordinates

We here introduce another variable change that transforms integrals from Cartesian to *spherical coordinates*. Such coordinates use a range r and angles θ_i, $i = 1, \dots, n-1$, to parametrize $\mathbb{R}^n$. For the two-dimensional case, these are merely the well known polar coordinates.

Without going into further details, we state that some of the integration rules of Section 4.4 are based on such a change. Among them is the rule the cubature Kalman filter by Arasaratnam and Haykin (2009) uses. The change facilitates that different rules can be chosen for the integration of the range and the spherical part, which gives rise to some flexibility in the integration techniques.

An important consequence is that functions $f(x)$ that merely depend on the range can be reduced to one-dimensional problems, which might even give rise to an analytical solution. Even without analytical solution, a one-dimensional integral can still be more efficiently solved than one over n dimensions. We return to this important point in Section 4.7.2.

The variable change

From the previous subsection, we know that expected values that involve elliptically contoured random vectors can be written as integrals with "standard" density. Therefore, we focus on these simplified integrals:

$$\mathrm{E}\left\{f(x)\right\} = \int g(y)h(y^{\mathrm{T}}y)\,\mathrm{d}y. \tag{4.21}$$

For the Gaussian case, we can identify $h(y^{\mathrm{T}}y) = \mathcal{N}(y; 0, I)$ and $g(y) = f(Uy + \hat{x})$.

We next introduce the coordinate transformation

$$\begin{aligned} y_1 &= r\sin(\theta_1), \\ y_2 &= r\cos(\theta_1)\sin(\theta_2), \\ y_3 &= r\cos(\theta_1)\cos(\theta_2)\sin(\theta_3), \\ &\vdots \\ y_{n-1} &= r\cos(\theta_1)\ldots\cos(\theta_{n-2})\sin(\theta_{n-1}), \\ y_n &= r\cos(\theta_1)\ldots\cos(\theta_{n-2})\cos(\theta_{n-1}). \end{aligned} \tag{4.22a}$$

The range and angles vary according to

$$r \in [0, \infty), \quad \theta_{n-1} \in [-\pi, \pi), \quad \theta_i \in [-\tfrac{\pi}{2}, \tfrac{\pi}{2}), \quad i = 1, \ldots, n-2, \tag{4.22b}$$

in order to parametrize $\mathbb{R}^n$. For $n = 2$ and $n = 3$, these transformations are essentially part of any undergraduate calculus course.

The Jacobian determinant of the variable change (4.22a) is

$$r^{n-1}\cos^{n-2}(\theta_1)\cos^{n-3}(\theta_2)\ldots\cos(\theta_{n-2}). \tag{4.23}$$

By applying the transformation theorem of Appendix A.1.4, with $h(y^{\mathrm{T}}y) = h(r^2)$, we derived the density in spherical coordinates:

$$p(r, \theta_1, \ldots, \theta_{n-1}) = r^{n-1}\cos^{n-2}(\theta_1)\cos^{n-3}(\theta_2)\ldots\cos(\theta_{n-2})h(r^2). \tag{4.24}$$

The radius and angles are independent because $p(r, \theta_1, \ldots, \theta_{n-1})$ can be factored as $p(r)p(\theta_1)\ldots p(\theta_{n-1})$.

Densities of range and squared range

The marginal density of r is obtained by integrating out all angles. With

$$C(n) = \int_{-\pi}^{\pi} \mathrm{d}\theta_{n-1} \int_{-\pi/2}^{\pi/2} \cos(\theta_{n-2})\,\mathrm{d}\theta_{n-2} \ldots \int_{-\pi/2}^{\pi/2} \cos^{n-2}(\theta_1)\,\mathrm{d}\theta_1 = \frac{2\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2})} \tag{4.25}$$

as the integral over the unit sphere, we have that

$$p(r) = C(n)r^{n-1}h(r^2), \quad r > 0. \tag{4.26}$$

Using the transformation theorem of Appendix A.1.4, we can also derive the density of $\rho = r^2$ from (4.26). The Jacobian determinant is here $\frac{1}{2}\rho^{-1/2}$ and

$$p(\rho) = C(n)\rho^{\frac{n-1}{2}} h(\rho)\frac{1}{2}\rho^{-1/2} = \frac{1}{2}C(n)h(\rho)\rho^{\frac{n}{2}-1}, \tag{4.27}$$

for $\rho > 0$. The squared range is interesting because it admits a well known distribution for important special cases.

Example 4.5: Squared range of a Gaussian random vector

For the Gaussian case, inserting the expressions (4.13) and (4.25) into (4.27) yields

$$\begin{aligned} p(\rho) &= \frac{1}{2}\frac{2\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2})}(2\pi)^{-\frac{n}{2}} \exp(-\frac{\rho}{2})\rho^{\frac{n}{2}-1} \\ &= \frac{2^{-\frac{n}{2}}}{\Gamma(\frac{n}{2})}\rho^{\frac{n}{2}-1}\exp(-\frac{1}{2}\rho) = \chi^2(\rho; n), \end{aligned} \tag{4.28}$$

for $\rho > 0$, which is a chi-squared density as given in (A.20) of Appendix A.2.2.

We have established that for a Gaussian random vector $x \sim \mathcal{N}(\hat{x}, P)$, the random variable $\Delta^2 = y^{\mathrm{T}}y = (x - \hat{x})^{\mathrm{T}}P^{-1}(x - \hat{x})$ admits a chi-squared distribution.

A related result can also be stated for the t distribution but requires the density of a scaled squared range $\zeta = r^2/n$:

$$p(\zeta) = \frac{1}{2}C(n)h(n\zeta)n^{\frac{n}{2}}\zeta^{\frac{n}{2}-1}, \tag{4.29}$$

for $\zeta > 0$.

Example 4.6: Scaled squared range of a *t* random vector

Repeating the above example with the corresponding equations for the t distribution, that is (4.14) and (4.25) in (4.29), we obtain

$$\begin{aligned} p(\zeta) &= \frac{1}{2}\frac{2\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2})}\frac{\Gamma(\frac{\nu+n}{2})}{\Gamma(\frac{\nu}{2})}\frac{1}{(\nu\pi)^{\frac{n}{2}}}\left(1 + \frac{1}{\nu}n\zeta\right)^{-\frac{n+\nu}{2}} n^{\frac{n}{2}}\zeta^{\frac{n}{2}-1} \\ &= \frac{\Gamma(\frac{\nu+n}{2})}{\Gamma(\frac{\nu}{2})\Gamma(\frac{n}{2})}\nu^{\frac{\nu}{2}}n^{\frac{n}{2}}\frac{\zeta^{\frac{n}{2}-1}}{(\nu + n\zeta)^{\frac{\nu+n}{2}}} = \mathcal{F}(\zeta; n, \nu). \end{aligned} \tag{4.30}$$

We have established that, for an n-dimensional $x \sim \mathrm{St}(\hat{x}, P, \nu)$, the scaled squared range $(x - \hat{x})^{\mathrm{T}}P^{-1}(x - \hat{x})/n$ admits an F-distribution $\mathcal{F}(n, \nu)$.

For functions $f(x)$ that merely depend on x via $(x - \hat{x})^{\mathrm{T}}P^{-1}(x - \hat{x})$, the above results for the Gaussian and t distribution are of importance, because they facilitate breaking the n-dimensional integrals in (4.3) down to scalar integrals. These can then be solved either analytically or via Monte Carlo methods.

4.3 Basic Monte Carlo Integration

This section describes *Monte Carlo integration* in its most basic form. For a more thorough treatment of Monte Carlo methods in general, the reader is referred to Robert and Casella (2004). Also, Bishop (2006) dedicates one chapter to sampling based methods.

We consider the central problem of computing the moments in (4.3). Moreover, we assume that we have the means to easily generate random realizations from the distribution of x. The central idea of Monte Carlo integration is to replace the expectation integrals by an average over a large number of transformed random realizations.

Let $\{x^{(i)}\}_{i=1}^{N}$ denote such an ensemble with $x^{(i)} \sim p(x)$. An approximation to (4.3a) is then given by

$$\hat{z} \approx \frac{1}{N} \sum_{i=1}^{N} f(x^{(i)}). \tag{4.31a}$$

The function f is evaluated at each sample $x^{(i)}$, and the outcomes are averaged. Justification for this scheme is given by the *law of large numbers*, which states that (4.31a) indeed converges to $\hat{z}$ as N tends to infinity.

In a similar manner, the covariance P_z can be computed:

$$P_z \approx \frac{1}{N} \sum_{i=1}^{N} \big(f(x^{(i)}) - \hat{z}\big)\big(f(x^{(i)}) - \hat{z}\big)^{\mathrm{T}}. \tag{4.31b}$$

Equation (4.31b) exists in two variants of which one is normalized by $(N-1)$ instead of N. The $(N-1)$ version yields an unbiased estimate of the covariance, whereas the N version corresponds to the maximum likelihood solution under the assumption that the transformed samples admit a Gaussian distribution. There is practically no difference for large N.

The values in (4.31) are never exact in the sense that there is always variation in the result, due to the random sampling. These variations decrease with increased sample size N, and can be reduced further by employing variance reduction techniques. Details are given in Robert and Casella (2004). Sometimes surprisingly accurate results can be obtained with only few samples.

One advantage of Monte Carlo methods is that we can treat arbitrary densities $p(x)$, as long as the corresponding random samples $x^{(i)}$ can be generated. If this is not the case, the concept of *importance sampling* can be used, which actually builds the basis for the particle filter of Section 3.3. More advanced schemes to draw samples from complicated distributions involve so called *Markov chain Monte Carlo* methods. The interested reader is referred to Robert and Casella (2004) for further details.

Furthermore, there are no restrictions on the functional form of f. If the expected values exist, then the method should work for any function. In contrast, the methods that we describe in Section 4.4 work only well if f is well approximated by a polynomial of certain degree.

A cautionary remark is that only expected values that in fact exist can be computed. In other words, if the involved integrals do not converge, the presented MC integration yields meaningless results. We demonstrate this using an example.

Example 4.7: "Cauchy mean"

The Cauchy distribution, which is a special case of the univariate t distribution, has the density

$$p(x) = \frac{1}{\pi\,(x^2+1)} \tag{4.32}$$

for location and scale parameters 0 and 1, respectively. Figure 4.1 shows that $p(x)$ appears bell shaped and centered around 0, with considerably heavier tails than a Gaussian density.

The Cauchy distribution has no defined mean, and no defined higher moments, and is in that sense an extreme example. While an infinite variance, for instance, can be intuitively explained, the reason for the lack of a defined mean involves some integration technicalities which we shall not review here.

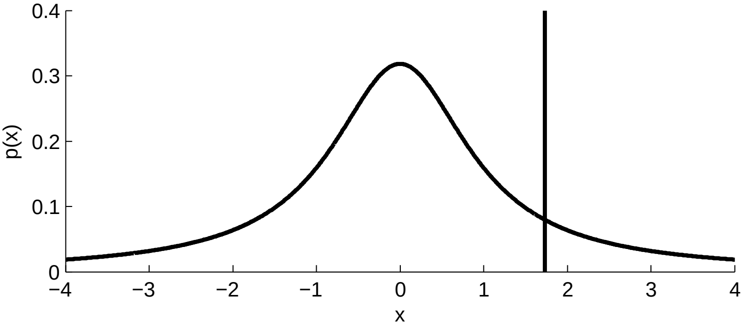


Figure 4.1: *The Cauchy density* (4.32). *The vertical line indicates a sample mean that is obtained by averaging the random samples in Figure 4.2.*

An interesting experiment is to nevertheless attempt to compute the "Cauchy mean" using the tools of this section. From the symmetry of $p(x)$, we might hope to recover the location parameter 0 using a sufficiently large N. Samples of one such experiment are illustrated in Figure 4.2. Obviously, some are very far from 0. Averaging the samples according to (4.31a) yields a value that is given by the vertical line in Figure 4.1, and differs considerably from 0, even for more than 10^5 random samples in this one-dimensional problem. Although most samples are concentrated around 0, the few extreme outliers heavily manipulate the averaging, and the results vary heavily from experiment to experiment. Hence, it is not possible to compute a meaningful "Cauchy mean" by Monte Carlo integration.

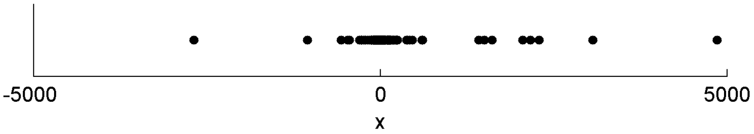


Figure 4.2: *Random samples from the Cauchy distribution with density* (4.32). *The sample mean gives the value that is indicated by the vertical line in Figure 4.1.*

4.4 Deterministic Integration Rules and the Unscented Transformation

We here present two schemes that appear very similar but are derived from different viewpoints: the *unscented transformation* and *deterministic integration rules*. Both rely on *deterministic sampling*, as opposed to the random sampling in Monte Carlo integration. More specifically, N points $x^{(i)}$ are chosen systematically and assigned weights $w^{(i)}$. The samples are transformed by the function f, and processed with the weights to solve the moment computation problem (4.3).

Each of the methods entails a specific sample and weight calculation rule. Common to all of them is, however, that the calculated mean and covariance are given as weighted sample statistics

$$\hat{z} \approx \sum_{i=1}^{N} w^{(i)} f(x^{(i)}), \tag{4.33a}$$

$$P_z \approx \sum_{i=1}^{N} w^{(i)} \big(f(x^{(i)}) - \hat{z}\big)\big(f(x^{(i)}) - \hat{z}\big)^{\mathrm{T}}, \tag{4.33b}$$

where the weights add up to one, $\sum_{i=1}^{N} w^{(i)} = 1$. A relation to the (weightless) sample statistics in (4.31) is given by the interpretation that a proportion $w^{(i)}$ of a large sample ensemble admits the value $x^{(i)}$ for each $i = 1, \dots, N$. In contrast to Monte Carlo methods, there is no variation in the results because the way in which the samples and weights in (4.33) are generated is fully deterministic. The sample size N is typically lower than in Monte Carlo integration, which makes the schemes computationally attractive.

4.4.1 The Unscented Transformation

The unscented transformation or UT was introduced in Julier et al. (1995) in a filtering context, as heart of the unscented Kalman filter or UKF. It is based on the intuitive idea of describing the distribution of a random variable x by a set of representative samples $\{x^{(i)}\}_{i=1}^N$, also called *sigma points*, and weights $\{w^{(i)}\}_{i=1}^N$. The weighted transformed samples $\{f(x^{(i)}), w^{(i)}\}_{i=1}^N$ should then ideally represent the distribution of $z = f(x)$ well, with moments (4.33) close to the true $\hat{z}$ and P_z.

The term UT refers to the suggested solution to the moment computation problem (4.3), and can be summarized as first generating N samples and weights, $\{x^{(i)}\}_{i=1}^N$ and $\{w^{(i)}\}_{i=1}^N$, transforming each $x^{(i)}$ to obtain $z^{(i)} = f(x^{(i)})$, and subsequently computing the weighted sample mean and covariance according to (4.33).

Such a scheme is appealing as it requires hardly more than evaluating f at the sigma points. Its implementation should not take more than a few lines of code. UT often performs well and gives exact results for linear f. However, its theoretical foundation is vague, and statements about the resulting accuracy are difficult to make for general f.

The samples $x^{(i)}$ and weights $w^{(i)}$ are chosen such that certain moments of x are preserved. Specifically, the weighted mean and covariance of $\{x^{(i)}\}_{i=1}^N$ should reflect $\hat{x}$ and P:

$$\textstyle\sum_{i=1}^N w^{(i)} x^{(i)} = \hat{x}, \qquad \sum_{i=1}^N w^{(i)} (x^{(i)} - \hat{x})(x^{(i)} - \hat{x})^{\mathrm{T}} = P. \tag{4.34}$$

There are infinitely many combinations of weights and samples that satisfy this condition. We present two schemes below.

The “standard” unscented transformation

The following selection and weighting scheme is taken from Julier et al. (2000) and utilizes a matrix square root U of P, similar to (4.16), that can be computed by *Cholesky factorization* or the *singular value decomposition*. We denote by u_i the i-th column of U. The $N = 2n + 1$ points and weights for an n-dimensional x are generated as follows:

$$x^{(0)} = \hat{x}, \tag{4.35a}$$

$$x^{(\pm i)} = \hat{x} \pm \sqrt{(n+\kappa)} u_i, \quad i = 1, \ldots, n, \tag{4.35b}$$

$$w^{(0)} = \frac{\kappa}{n+\kappa}, \tag{4.35c}$$

$$w^{(\pm i)} = \frac{1}{2(n+\kappa)}, \quad i = 1, \ldots, n. \tag{4.35d}$$

We here changed notation and used the superscript $(\pm i)$ to emphasize the symmetry in the sample set. Accordingly, the summation in (4.33) should be replaced by a sum from $-n$ to n. Moreover, using the transformed samples

$$z^{(0)} = f(x^{(0)}), \quad z^{(\pm i)} = f(x^{(\pm i)}), \tag{4.36}$$

we can re-formulate the UT approximation to (4.3a) as

$$\hat{z} \approx w^{(0)} z^{(0)} + \sum_{i=1}^{n} w^{(\pm i)} \left(z^{(+i)} + z^{(-i)} \right). \tag{4.37}$$

As we shall see, (4.37) has a connection to the moment computation schemes in Sections 4.5 and 4.6.

The central point is discarded for $\kappa = 0$, which yields $2n$ equally weighted samples. The UT is then identical to a cubature rule of degree 3, that is presented in Section 4.4.2.

The scaled unscented transformation

For larger n, the points $x^{(\pm i)}$ in (4.35) are further from the mean $\hat{x}$. These expansive samples might yield degraded results for certain functions f that change severely over the x-space. Therefore, Julier (2002) introduced a "scaled" UT version. The points in (4.35) are modified without altering the statistics in (4.34). Julier (2002) applies a two step procedure that has been condensed to a selection scheme similar to (4.35) by van der Merwe (2004).

Specifically, the parameter κ is replaced by $\lambda = \alpha^2(n+\kappa) - n$. Furthermore, another scalar parameter β is introduced. The samples and weights are given by

$$\begin{aligned}
x^{(0)} &= \hat{x}, && \text{(4.38a)}\\
x^{(\pm i)} &= \hat{x} \pm \sqrt{n+\lambda}\, u_i = \hat{x} \pm \alpha \sqrt{n+\kappa}\, u_i, \quad i = 1, \ldots, n, && \text{(4.38b)}\\
w_m^{(0)} &= \frac{\lambda}{n+\lambda}, && \text{(4.38c)}\\
w_c^{(0)} &= \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta, && \text{(4.38d)}\\
w^{(\pm i)} &= \frac{1}{2(n+\lambda)}, \quad i = 1, \ldots, n. && \text{(4.38e)}
\end{aligned}$$

Here, (4.38b) reveals that α can be used to scale the spread of samples. (4.38c) and (4.38d) are two different weights for the central (0)-terms in the mean and covariance calculations (4.33a) and (4.33b), respectively. The inclusion of β minimizes the deviation between true covariance P_z and its approximation, and should be chosen as $\beta = 2$ for a Gaussian x. With considerable effort, the details can be extracted from Julier and Uhlmann (2004). The interested reader is also referred to van der Merwe (2004) for further insights and a guideline on how to choose the parameters in (4.38). Essentially, κ becomes obsolete, $0 < \alpha \leq 1$ should depend on the specific function f as well as $\operatorname{cov}\{x\} = P$.

More advanced sample and weight selection schemes than (4.38) are conceivable. These might not necessarily build on the concept of *moment matching*, as seen in Hanebeck et al. (2009).

A cautionary remark

The intuitive idea behind UT is also its weak point. The weighted samples might reflect the random variable x very well, have the same mean and covariance, and also capture some higher moments. However, very little can be said about the transformed samples $\{z^{(i)}\}_{i=1}^{N}$. How the characteristics of the weighted sample set carries over is highly dependent on the function f.

With the EKF as main competitor to UKF at the time of its introduction, Julier et al. (2000) showed that UT is better than linearization, which is obviously true. However, some accuracy statements that can be found in literature should be interpreted with care. We continue this discussion in Section 4.4.3. Still, UT often yields good results and enjoys great popularity due to its simple implementation.

4.4.2 Deterministic Integration Rules

The deterministic integration rules of this section were initially developed to solve one-dimensional integrals of an exponentially weighted function $g(y)$ over the entire real line,

$$\int_{-\infty}^{\infty} g(y)\exp(-y^2)\,\mathrm{d}y. \tag{4.39}$$

The problem is extended to n dimensions by understanding (4.39) as n-fold integral and replacing the weight,

$$\int\!...\!\int g(y)\exp(-y^{\mathrm{T}}y)\,\mathrm{d}y_1...\,\mathrm{d}y_n. \tag{4.40}$$

The exponential weight establishes a connection to the Gaussian distribution, because any expected value (4.3a) with $x \sim \mathcal{N}(\hat{x}, P)$ can be written as (4.40). As a consequence of the defined structure in the integral (4.40), we here treat the moment computation problem (4.3) for Gaussian random variables only.

Problems such as (4.39) and (4.40) have been addressed by mathematicians for centuries. The suggested solutions, for example *Gauss-Hermite quadrature* rules, compute the integrals by summing weighted function evaluations, similar to (4.33a),

$$\int g(y)\exp(-y^{\mathrm{T}}y)\,\mathrm{d}y \approx \textstyle\sum_{i=1}^{N} w^{(i)} g(y^{(i)}). \tag{4.41}$$

The scheme (4.41) exact for polynomial functions $g(y)$ of a certain degree, which depends on the number of used points N and the chosen samples $y^{(i)}$ and weights $w^{(i)}$. The relevant literature is hardly new, with all of the material essentially covered in the monograph Stroud (1971). Of course, the field of *numerical integration* has developed since then. We stick to "classical" methods because they fit the framework of this section. After all, (4.3) is solved using expressions similar to (4.33) here.

Stroud (1971) groups integration rules into two families. *Product rules* treat each dimension in an n-fold integral of the form (4.40) separately. This typically results in a computational demand that is prohibitive in high dimensions, as we shall see. *Nonproduct rules* take into account all dimensions at once, and are usually cheaper to compute. Each family is treated in a separate section below. Admittedly, the monograph Stroud (1971) is out of print and hard to get hold of. References to easier to find sources are given where available.

In the signal processing community, the here presented numerical integration techniques have gained attention because of their use in Kalman filters. The computationally demanding but accurate *Gauss-Hermite Kalman filter* by Ito and Xiong (2000) and the *cubature Kalman filter* (CKF) by Arasaratnam and Haykin (2009) are two examples. The former employs a product rule, whereas the latter is based on a nonproduct rule. Several filters, including UKF, are analyzed from a numerical integration viewpoint in Wu et al. (2006), which opens up for assessing not only accuracy but also numerical properties. Recently, Jia et al. (2013) employed nonproduct rules that are more accurate than those in CKF, but also require more samples.

Deterministic integration rules are not only attractive because they offer a simple structure and an assessable computational cost. More important is that each integration rule comes with an accuracy statement: an integration rule of *degree* d solves the integral (4.40) exactly for *polynomial* functions $g(y)$ with degree d or lower. Here, the degree of a polynomial function $g(y)$ refers to the highest appearing *monomial* degree. For example, the function $f(y) = [y_1^3 y_2^2 - 3,\ y_2]^{\mathrm{T}}$ has degree 5 because of the degree-5 monomial $y_1^3 y_2^2$.

From Gaussian expected values to integrals with exponential weight

We next relate the expected values (4.3) to the integral with exponential weight (4.40). This is achieved via a change of integration variables. The treatment applies only to Gaussian random variables $x \sim \mathcal{N}(\hat{x}, P)$.

The following procedure is a slight variation of the material in Section 4.2.1. We consider the variable change relation

$$x = \sqrt{2}Uy + \hat{x}, \tag{4.42}$$

where U is a matrix square root of the covariance matrix P, similar to (4.16). The Jacobian of (4.42) is $\sqrt{2}U$, and $\det(\sqrt{2}U) = 2^{\frac{n}{2}} \det(U)$. With a Gaussian density, (4.3a) can be written as

$$\int f(x)\mathcal{N}(x; \hat{x}, P)\,\mathrm{d}x = \int \underbrace{\frac{1}{\pi^{\frac{n}{2}}} f(\sqrt{2}Uy + \hat{x})}_{g(y)} \exp(-y^{\mathrm{T}}y)\,\mathrm{d}y, \tag{4.43}$$

which is of the desired form (4.40).

Product rules

Product rules are essentially based on computing the n-fold integral (4.40) by treating one dimension at a time. We therefore review a popular technique for scalar problems (4.39) first.

Gauss-Hermite quadrature is a scheme that uses m points $y^{(i)}$ and weights $w^{(i)}$ according to (4.41), and integrates polynomials of degree $d = 2m - 1$ exactly. The points $y^{(i)}$ are roots of an *Hermite polynomial* that is determined by m. Also the weights $w^{(i)}$ are calculated from such a polynomial. The details are listed in Chapter 25 of Abramowitz and Stegun (1965).

Applying a product rule to an n-fold integral works as follows. First, all components but y_1 are ignored. If an m-point Gauss-Hermite rule is applied, then the solved y_1-integral is actually a sum of m functions, one term for each $y_1^{(i)}$. Next, an m-point rule for y_2 is applied, which yields a sum of m^2 functions, one term for each pair $(y_1^{(i)}, y_2^{(j)})$, and so on.

In summary, applying a product rule based on scalar m-point rules for an n-fold integral requires $N = m^n$ points in total. Although the overall accuracy can be high with the rule having degree $d = 2m - 1$, the sheer amount of required function evaluations renders the application difficult for high-dimensional problems.

Nonproduct rules

Among the family of nonproduct rules for weighted integrals (4.40) is the degree-3 cubature rule that is used in the CKF of Section 3.2. Arasaratnam and Haykin (2009) provide the rule and also its derivation, but also earlier sources, Stroud (1971) and Stroud and Secrest (1963), contain the material.

We next present the rule and relate it to the UT of Section 4.4.1. We also adopt the $(\pm i)$ notation of (4.35) again. The degree-3 rule for the integration problem (4.40) requires $2n$ points and weights that are generated according to

$$y^{(\pm i)} = \sqrt{\frac{n}{2}} e_i, \tag{4.44a}$$

$$w^{(\pm i)} = \frac{\pi^{\frac{n}{2}}}{2n}, \quad i = 1, \ldots, n, \tag{4.44b}$$

where e_i denotes the i-th unit vector (the i-th column of the identity matrix). We can translate the $y^{(\pm i)}$ back to $x^{(\pm i)}$ using the variable change (4.42). The obtained

$$x^{(\pm i)} = \hat{x} \pm \sqrt{2} U y^{(\pm i)} = \hat{x} \pm \sqrt{n} u_i, \quad i = 1, \ldots, n, \tag{4.45}$$

are the same points as in the UT selection scheme (4.35b) for $\kappa = 0$.

The integration rule (4.41) requires processing weighted transformed samples

$$w^{(\pm i)} g(y^{(\pm i)}) = \frac{\pi^{\frac{n}{2}}}{2n} \frac{1}{\pi^{\frac{n}{2}}} f(x^{(\pm i)}) = \frac{1}{2n} f(x^{(\pm i)}), \tag{4.46}$$

which coincide with the weighted samples in an UT scheme (4.35) with $\kappa = 0$. We have thus shown the equivalence between the degree-3 integration rule and UT for one choice of parameters.

Stroud (1971) lists several nonproduct rules for solving (4.40), including the above degree-3 rule with $2n$ points, a degree-5 rule with $2n^2 + 1$ points, and a degree-7 rule with $(4n^3 + 8n + 3)/3$ points. Some of the results are also given in Stroud and Secrest (1963) and Stroud (1967).

4.4.3 Accuracy Statements

The purpose of the following paragraphs is to summarize accuracy statements about the unscented transformation and deterministic integration rules. Hopefully this can help avoid misconceptions about what the presented moment computation schemes are capable of, and what they cannot achieve.

For the sake of clarity, we reserve the term "degree" for the degree of a polynomial or monomial, and the term "order" for the derivative of a certain order. Statistical moments are written with neither "degree" nor "order", as in "first moment".

The paper Julier et al. (2000) contains several statements involving these terms. First, the first and second moments of x, that is the mean and covariance of x, are preserved in the weighted samples $\{w^{(i)}, x^{(i)}\}_{i=1}^{N}$. Second, based on an infinite Taylor series of $z = f(x)$, an infinite series for P_z of (4.3b) is formulated. A linearization scheme provides only a truncated version of the true P_z. It is claimed that UT does not involve truncation and that, hence, the computed covariance is "correct up to the second order." Whether this "order" refers to degree of the Taylor polynomial of P_z, or to the moments of the transformed samples, is difficult to say. Unfortunately, this statement is easily misinterpreted, and can raise expectations that UT cannot live up to. For instance, UT certainly does not give the correct mean $\hat{z}$ and covariance P_z for arbitrary nonlinearities. Also, UT does not give the same results as a scheme that approximates $f(x)$ by a degree-2 Taylor polynomial, as we will demonstrate in Section 4.7.1. Still, UT often works well. Concise accuracy statements, however, are not available in general.

The integration perspective that we have assumed in Section 4.4.2 offers immediate accuracy statements: a rule of degree d integrates exponentially weighted polynomials with degree d or lower exactly. That, in turn, means that we can compute expected values (4.3a) with respect to Gaussian densities exactly, if the function $f(x)$ is a polynomial of degree d or lower. That solves half of the moment computation problem. The covariance computations (4.3b) require expected values of the squared function, which might not be exact for the chosen rule. We demonstrate this in an example.

Example 4.8: Polynomial function and degree-3 integration rule

Consider the two-dimensional $x \sim \mathcal{N}(\hat{x}, P)$ and the transformed random variable z, given by

$$z = f(x) = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{12} & b_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} x_1^2 \\ 0 \end{bmatrix}. \tag{4.47}$$

Each component of $f(x)$ is a polynomial in x_1 and x_2. The degree of $f(x)$ is 2.

Now, suppose that we apply an integration rule of degree 3 to compute the moments (4.3). Then, $\hat{z}$ should be computed correctly because it is a polynomial of degree 2. The computation of P_z involves integration of the squared function. The entries in

$$P_z = \begin{bmatrix} P_{z,11} & P_{z,12} \\ P_{z,12} & P_{z,22} \end{bmatrix} \tag{4.48}$$

require integrating polynomials of degree 4 ($P_{z,11}$), degree 3 ($P_{z,12}$), and degree 2 ($P_{z,22}$). Hence, we can expect the degree-3 rule to compute all entries but $P_{z,11}$ correctly.

Still, the errors of applying a degree-d rule to a polynomial of degree $d+1$ or higher are difficult to assess.

A user that wants to make sure that a certain integration rule performs well must know about the structure of the nonlinearity. This establishes a connection to the Taylor series of the next chapter, which can be employed to analyze nonlinear functions.

4.5 Truncated Taylor Series and Related Nonlinearity Approximations

In contrast to the previously mentioned methods, which take the involved functions as they are, we now work with approximations of $f(x)$. In that respect, we are looking for exact solutions to problems that are simpler than (4.3), but hopefully sufficiently similar to (4.3).

A simple approximation is the linearization of $f(x)$, which can be achieved using a number of techniques. Whether we apply a linear interpolation to f, truncate the Taylor series of f after the linear term, or optimize some criterion in order to find the parameters, the result of a linearization is always of the form

$$f(x) \approx Ax + b, \tag{4.49}$$

and simplifies the moment calculations (4.3) significantly:

$$\mathrm{E}\{Ax+b\} = A\hat{x} + b, \tag{4.50a}$$

$$\mathrm{cov}\{Ax+b\} = APA^{\mathrm{T}}, \tag{4.50b}$$

for x with mean $\hat{x}$ and covariance P. The simplicity of these equations is deceiving because a linearized system is not able to capture relevant characteristics of $f(x)$ for many functions. As a consequence, linearization based approaches might not work well. In a filtering context, the linearization based EKF of Section 3.2 diverges for certain applications. The reason is often an oversimplification of the underlying system. If a linearization is carried out, it should be carefully checked whether it is valid.

In the course of this section, we look into the details of several approaches, including and beyond linearization. We begin by re-formulating the multivariable Taylor series of functions $f(x)$, which facilitates expressions as compact as (4.50) for the moment computation problem (4.3). The Taylor series requires computing derivatives, a topic we review as well. The section is concluded by interpolation or divided difference approaches for approximating $f(x)$.

4.5.1 A Convenient Formulation of the Taylor Series

The Taylor series, as introduced in any undergraduate calculus course, can be formulated for differentiable functions $f(x)$. Because we consider n-dimensional vectors x, the number of partial derivatives that need to be computed for the degree-d Taylor polynomial increases quickly. Also, the notation becomes cumbersome. We reformulate the Taylor series using *derivative operators* and the *Kronecker product*, and obtain compact expressions that very much simplify mean and covariance calculations of the degree-d Taylor polynomials.

A related approach, with a focus on degree-2 Taylor polynomials, is presented in Manolakis (2011).

The Taylor series as infinite sum

We first consider a scalar function $f : \mathbb{R}^n \to \mathbb{R}$. Under the assumptions that the partial derivatives exist, $f(x)$ can be written as infinite series about an expansion point $\bar{x}$. Using the deviation vector $\widetilde{x} = x - \bar{x}$, the Taylor series is given by

$$\begin{aligned} f(x) = f(\bar{x}) &+ \sum_{i=1}^{n} \frac{\partial}{\partial x_i} f(\bar{x}) \widetilde{x}_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2}{\partial x_i \partial x_j} f(\bar{x}) \widetilde{x}_i \widetilde{x}_j \\ &+ \frac{1}{3!} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial^3}{\partial x_i \partial x_j \partial x_k} f(\bar{x}) \widetilde{x}_i \widetilde{x}_j \widetilde{x}_k + \dots . \end{aligned} \quad (4.51)$$

Here, $\frac{\partial}{\partial x_i} f(\bar{x})$ denotes the partial derivative of $f(x)$ with respect to x_i, that is subsequently evaluated at $\bar{x}$.

The infinite series (4.51) is a polynomial in the components of the deviation vector $\widetilde{x}$. If we assume that all partial derivatives are available, it is apparent that we can compute $\mathrm{E}\{f(x)\}$ if we know the expected values

$$\mathrm{E}\{\widetilde{x}_i\}, \mathrm{E}\{\widetilde{x}_i \widetilde{x}_j\}, \mathrm{E}\{\widetilde{x}_i \widetilde{x}_j \widetilde{x}_k\}, \dots \quad (4.52)$$

for all combinations of $i = 1, ..., n$; $j = 1, ..., n$; $k = 1, ..., n$; and so on. The quantities are *moments* in $\widetilde{x}_i$ and become *central moments* in x for $\bar{x} = \hat{x}$. Expressions for such moments are available for many distributions, for instance the Gaussian distribution, as we will see in Example 4.9. We have established that, in principle, $\mathrm{E}\{f(x)\}$ can be computed up to arbitrary accuracy if the required partial derivatives and moments are available. That solves half of the moment computation problem (4.3). A specific formula for a scalar Gaussian x is given in Saha et al. (2009).

The next step is to consider two functions $f_1 : \mathbb{R}^n \to \mathbb{R}$ and $f_2 : \mathbb{R}^n \to \mathbb{R}$. Even though we provided a conceptual solution to compute $\mathrm{E}\{f_1(x)\}$ and $\mathrm{E}\{f_2(x)\}$ up to arbitrary accuracy, the prospect of computing $\operatorname{cov}\{f_1(x), f_2(x)\}$ involves multiplying infinite sums and appears thus cumbersome and daunting.

Derivative operators and Kronecker products

In order to overcome the shortcomings of (4.51), we need to come up with tools that simplify the expressions. First, we introduce the *derivative operator* D, that acts in the following manner

$$Df(x) = \left[\tfrac{\partial}{\partial x_1} f(x), \quad \tfrac{\partial}{\partial x_2} f(x), \quad \dots, \quad \tfrac{\partial}{\partial x_n} f(x)\right]. \tag{4.53}$$

For a scalar $f(x)$, $Df(x)$ is a row vector. However, we can introduce the convention that D operates on each row $f_l(x)$ of a function $f(x)$ separately. Then, $Df(x)$ yields the Jacobian matrix of $f(x)$ for a vector valued function.

If we ignore the operator characteristics and treat D as regular vector, we can apply the Kronecker product, as in Van Loan (2000), to D:

$$\begin{aligned} D \otimes D &= \left[\frac{\partial}{\partial x_1}, \quad \frac{\partial}{\partial x_2}, \quad \dots, \quad \frac{\partial}{\partial x_n}\right] \otimes \left[\frac{\partial}{\partial x_1}, \quad \frac{\partial}{\partial x_2}, \quad \dots, \quad \frac{\partial}{\partial x_n}\right] \\ &= \left[\frac{\partial^2}{\partial x_1^2}, \quad \frac{\partial^2}{\partial x_1 \partial x_2}, \quad \dots, \quad \frac{\partial^2}{\partial x_n \partial x_{n-1}}, \quad \frac{\partial^2}{\partial x_n^2}\right]. \end{aligned} \tag{4.54}$$

Going one step further, we define

$$D^{\{d\}} = D \otimes D \otimes \ldots \otimes D \tag{4.55}$$

as d-fold Kronecker product of D. The operator in (4.55) provides the means to compactly collect all partial derivatives of order d for a function $f(x)$. For example,

$$D^{\{2\}} f(x) = (D \otimes D) f(x) = \left[\tfrac{\partial^2 f(x)}{\partial x_1 \partial x_1}, \quad \tfrac{\partial^2 f(x)}{\partial x_1 \partial x_2}, \quad \tfrac{\partial^2 f(x)}{\partial x_2 \partial x_1}, \quad \tfrac{\partial^2 f(x)}{\partial x_2 \partial x_2}\right] \tag{4.56}$$

contains all second order partial derivatives of $f(x)$ with a two-dimensional x. For a scalar $f(x)$, these are all entries in the *Hessian* matrix of $f(x)$.

In a similar manner, repeated application of the Kronecker product to $\tilde{x}$ yields a vector that contains all monomials of a certain degree d. An example for $d = 2$ and $n = 2$ is given by

$$\tilde{x}^{\{2\}} = \tilde{x} \otimes \tilde{x} = \left[\tilde{x}_1^2, \quad \tilde{x}_1 \tilde{x}_2, \quad \tilde{x}_2 \tilde{x}_1, \quad \tilde{x}_2^2\right]^{\mathrm{T}}, \tag{4.57}$$

and contains all monomials of degree two.

Both (4.56) and (4.57) contain repeated entries and may thus seem unnecessarily blown up. However, the simple structure facilitates generating $\tilde{x}^{\{d\}}$ recursively from $\tilde{x}^{\{d-1\}}$. In a similar manner, $D^{\{d\}} f(x)$ can be obtained by differentiation of the entries in $D^{\{d-1\}} f(x)$. Here, the full capabilities of *computer algebra systems* such as Mathematica can be exploited. The entries in $D^{\{d\}} f(x)$ can be computed one scalar component at a time, and intermediate results can be simplified automatically. Repeated computation of derivatives can be avoided with some effort.

Compact Taylor polynomials

Multiplication of the vectors in (4.56), evaluated at $\overline{x}$, and (4.57) yields all degree-2 terms in the Taylor series of $f(x)$ with two-dimensional x. That is, we obtain the double sum in (4.51). More general, the product

$$D^{\{d\}}f(x)\widetilde{x}^{\{d\}} \tag{4.58}$$

gives all the degree-d terms in the Taylor series, except for a factor $1/d!$.

Using this insight, we can rewrite (4.51) as follows:

$$\begin{aligned} f(x) &= f(\overline{x}) + Df(\overline{x})\widetilde{x} + \frac{1}{2}D^{\{2\}}f(\overline{x})\widetilde{x}^{\{2\}} + \ldots + \frac{1}{d!}D^{\{d\}}f(\overline{x})\widetilde{x}^{\{d\}} + \ldots \\ &= \underbrace{\left[f(\overline{x}),\quad D^{\{1\}}f(\overline{x}),\quad \ldots,\quad \tfrac{1}{d!}D^{\{d\}}f(\overline{x})\right]}_{\mathbf{F}} \underbrace{\begin{bmatrix}1\\ \widetilde{x}^{\{1\}}\\ \vdots \\ \widetilde{x}^{\{d\}}\end{bmatrix}}_{\widetilde{\mathbf{x}}} + \ldots \end{aligned} \tag{4.59}$$

We managed to write the degree-d Taylor polynomial as high-dimensional linear relation in the monomial vector $\widetilde{\mathbf{x}}$.

With the convention that D acts on each row of a function separately, (4.59) applies to vector valued functions $f(x)$ without further change. This is already one advantage over the common notation to express the Taylor series of a scalar function with Jacobian and Hessian matrix, as in

$$f(x) = f(\overline{x}) + J(\overline{x})\widetilde{x} + \frac{1}{2}\widetilde{x}^{\mathrm{T}}H(\overline{x})\widetilde{x} + \ldots, \tag{4.60}$$

with $J(\overline{x})$ and $H(\overline{x})$ that contain first and second order partial derivatives of $f(x)$, respectively. Not only is (4.60) restricted to scalar functions, it can also only illustrate terms up to degree two without significant effort.

Of course, the compact notation in (4.59) conceals that the computation of $\mathbf{F}$ requires a lot of partial derivatives . For an n-vector x and the degree d Taylor polynomial, $(n+d)!/(d!n!) - 1$ unique partial derivatives are required, for example 20 for $d = 2, n = 5$, or 125 for $d = 4, n = 5$. The sheer amount of derivatives to be computed is one good reason to employ computer algebra systems, for example Mathematica.

Nonlinearity analysis using the Taylor series

In (4.59), we present a compact formulation of the degree-d Taylor polynomial of $f(x)$. Using software tools such as Mathematica, the provided expression can be used to analyze a function $f(x)$ with regard to its nonlinearity. The main idea is to compare the discrepancy between $f(x)$ and its degree-d polynomial approximation

$$\|f(x) - \mathbf{F}\widetilde{\mathbf{x}}\|, \tag{4.61}$$

where any norm may be chosen. Furthermore, (4.61) requires an expansion point $\bar{x}$ and a value x, at which the discrepancy is evaluated. For low-dimensional x, Mathematica's `Manipulate` function can be used to vary $\bar{x}$ and x, and simultaneously evaluate (4.61) for different degrees d. Also, an investigation of random samples $x \sim \mathcal{N}(\hat{x}, P)$ is conceivable with $\bar{x} = \hat{x}$. If (4.61) is small for $d = 1$ already, regardless of x and $\bar{x}$, then a linearization of $f(x)$ seems promising.

4.5.2 Expected Values of the Degree-d Taylor Polynomial

The main advantage of the linear relation in the monomial vector $\tilde{\mathbf{x}}$ in (4.59) is that it can be used to solve moment computation problem (4.3).

For many distributions, in particular the Gaussian distribution, the statistics of $\tilde{\mathbf{x}}$ are known. For $x \sim \mathcal{N}(\hat{x}, P)$, and $\bar{x} = \hat{x}$ as expansion point, the monomial vector $\tilde{\mathbf{x}}$ contains *central moments* that are readily available in functional form. Consequently, it is possible to compute $\mathrm{E}\{\tilde{\mathbf{x}}\}$ and $\mathrm{cov}\{\tilde{\mathbf{x}}\}$.

Example 4.9: Gaussian x

For a two-dimensional Gaussian random vector x, that is $n = 2$, the degree-2 monomial vector, that is $d = 2$, is given by

$$\tilde{\mathbf{x}} = \begin{bmatrix} 1, & \tilde{x}_1, & \tilde{x}_2, & \tilde{x}_1^2, & \tilde{x}_1\tilde{x}_2, & \tilde{x}_1\tilde{x}_2, & \tilde{x}_2^2 \end{bmatrix}^{\mathrm{T}}. \tag{4.62a}$$

The expected value of the monomial vector is

$$\mathrm{E}\{\tilde{\mathbf{x}}\} = \begin{bmatrix} 1, & 0, & 0, & P_{11}, & P_{12}, & P_{12}, & P_{22} \end{bmatrix}^{\mathrm{T}}, \tag{4.62b}$$

and the covariance can be computed as follows:

$$\mathrm{cov}\{\tilde{\mathbf{x}}\} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & P_{11} & P_{12} & 0 & 0 & 0 & 0 \\ 0 & P_{12} & P_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2P_{11}^2 & 2P_{11}P_{12} & 2P_{11}P_{12} & 2P_{12}^2 \\ 0 & 0 & 0 & 2P_{11}P_{12} & P_{12}^2 + P_{11}P_{22} & P_{12}^2 + P_{11}P_{22} & 2P_{12}P_{22} \\ 0 & 0 & 0 & 2P_{11}P_{12} & P_{12}^2 + P_{11}P_{22} & P_{12}^2 + P_{11}P_{22} & 2P_{12}P_{22} \\ 0 & 0 & 0 & 2P_{12}^2 & 2P_{12}P_{22} & 2P_{12}P_{22} & 2P_{22}^2 \end{bmatrix}. \tag{4.62c}$$

Obviously, $\mathrm{cov}\{\tilde{\mathbf{x}}\}$ contains the original $\mathrm{cov}\{x\} = P$ as block matrix. Furthermore, the matrix contains many zeros because many odd moments are zero for the symmetric $\tilde{x}$.

Equipped with the statistics of the monomial vector and the matrix $\mathbf{F}$, the mean and covariance of the degree-d Taylor polynomial of $f(x)$ are obtained by

$$\mathrm{E}\{\mathbf{F}\tilde{\mathbf{x}}\} = \mathbf{F}\,\mathrm{E}\{\tilde{\mathbf{x}}\}, \tag{4.63a}$$

$$\mathrm{cov}\{\mathbf{F}\tilde{\mathbf{x}}\} = \mathbf{F}\,\mathrm{cov}\{\tilde{\mathbf{x}}\}\,\mathbf{F}^{\mathrm{T}}. \tag{4.63b}$$

The relations to the linearized expected values (4.50) are apparent, and obtained as special case for $d = 1$.

Also, cross-covariances between $f(x)$ and x can be computed via

$$\operatorname{cov}\{x, f(x)\} \approx \operatorname{cov}\{x, \mathbf{F}\widetilde{\mathbf{x}}\} = \operatorname{cov}\{x, \widetilde{\mathbf{x}}\}\,\mathbf{F}^{\mathrm{T}}. \tag{4.64}$$

An extension to covariance computations for two vector valued functions $f_1(x)$ and $f_2(x)$ is straightforward. Using degree-d_1 and degree-d_2 Taylor polynomials with expansion point $\hat{x}$ to approximate $f_1(x)$ and $f_2(x)$, we can formulate

$$\mathbf{F}_l = \left[f_l(\overline{x}), \quad \tfrac{1}{2}D^{\{1\}}f_l(\overline{x}), \quad \ldots, \quad \tfrac{1}{d_l!}D^{\{d_l\}}f_l(\overline{x})\right], \tag{4.65a}$$

$$\widetilde{\mathbf{x}}_l = \left[1, \quad \widetilde{x}^{\{1\}}, \quad \ldots, \quad \widetilde{x}^{\{d_l\}}\right], \quad l = 1, 2. \tag{4.65b}$$

The covariance between $f_1(x)$ and $f_2(x)$ is then simply

$$\operatorname{cov}\{f_1(x), f_2(x)\} = \mathbf{F}_1 \operatorname{cov}\{\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2\}\,\mathbf{F}_2^{\mathrm{T}}. \tag{4.66}$$

Here, the subscripts denote which function is meant, and not the component of a vector.

4.5.3 A Cautionary Remark on Function Approximation

The following discussion is motivated by the use of Taylor polynomials to compute expected values of functions $f(x)$, but also applies to any other function approximation technique.

Truncated Taylor series, that is Taylor polynomials of degree d, can be used to approximate functions $f(x)$ as long as the required partial derivatives exist and can be computed. The approximation is a polynomial in the deviation from the expansion point $\widetilde{x} = x - \overline{x}$. Close to the expansion point $\overline{x}$, such an approximation is usually very good, and becomes better for higher degrees d. However, due to the dependence on degree-d monomials of the components of $\widetilde{x}$, the approximation error can grow quickly with increasing distance to $\overline{x}$. The user should be aware of this.

In particular, caution is advised if a function is approximated using a Taylor polynomial to facilitate the computation of expected values. This is because a local approximation is here used to facilitate computing a global quantity. After all, expected values usually require integration over the entire space. Luckily, expected values

$$\int f(x)p(x)\,\mathrm{d}x \tag{4.67}$$

include a weight $p(x)$ which turns the integration over the entire space into a somewhat local problem again. Any approximation to $f(x)$ should ideally be valid where x has most of its probability mass. For Gaussian vectors $x \sim \mathcal{N}(\hat{x}, P)$, the covariance information should therefore be used to see whether, for instance, a Taylor polynomial of degree d is reasonable.

4.5.4 Remarks on Derivative Computations

In order to advocate for the use of Taylor polynomials, we must make some statement on how to practically compute the involved partial derivatives. After all, it seems that computing derivatives is deemed as cumbersome, prone to errors, and impossible to implement. These concerns are here addressed by presenting three different derivative computation methods beyond pen and paper.

A related discussion, motivated by the use of derivatives in a completely different field, is given in Nocedal and Wright (2006).

Symbolic computations

Computer algebra systems or CAS such as Mathematica or Maple come with an immense inventory of functions. If a function is listed in Abramowitz and Stegun (1965), it is also implemented in modern symbolic software. Moreover, in contrast to the average user, CAS can manipulate and simplify complicated expressions automatically. When it comes to differentiation of analytical expressions, it is advised to at least verify all pen and paper calculations using a CAS.

With an increased functionality, including probability theoretic tools, modern CAS provide the means to approach many signal processing problems. Furthermore, a thorough a-priori analysis of a problem can sometimes save pages of MATLAB code.

Automatic differentiation

Algorithmic differentiation or AD, as presented in Griewank and Walther (2008), is a technique that automatically differentiates functions without making any approximations. The idea is based on the fact that each function is composed of elemental subfunctions that are simple to differentiate. Derivatives can thus be computed by repeated use of the chain rule, in a manner that some reader might recognize from the back-propagation technique in neural networks, Bishop (2006). AD algorithms take a function as input, and provide the user with another function that implements the desired derivative.

AD tools for a number of different programming languages, including MATLAB, are listed on the dedicated web page `www.autodiff.org`.

Divided finite differences

An obvious idea is to replace derivatives with finite divided differences. For numerical reasons, this is only recommended for first and second order partial derivatives.

The first order partial derivative of $f(x)$ with respect to the component x_i is in a *forward difference* scheme given by

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x + te_i) - f(x)}{t}. \tag{4.68a}$$

The function is perturbed by te_i, where t is a scalar parameter and e_i a unit vector with one as i-th component. Due to the finite accuracy in evaluating f, the parameter t must not be chosen too small, because otherwise the obtained difference is zero. On the other hand, choosing a large t might not reflect the derivative well either. Nocedal and Wright (2006) provides a thorough discussion of this, and concludes that t must be determined by the *unit roundoff* of the utilized computer.

Differencing once again yields the second order partial derivative

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} \approx \frac{f(x + te_i + te_j) - f(x + te_i) - f(x + te_j) + f(x)}{t^2}. \tag{4.68b}$$

Here, the choice of t becomes more critical with a division by t^2 that must yield a meaningful result.

More accurate partial derivatives can be computed using divided *central differences*:

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x + te_i) - f(x - te_i)}{2t}, \tag{4.69a}$$

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} \approx \frac{f(x+te_i+te_j) - f(x-te_i+te_j) - f(x+te_i-te_j) + f(x-te_i-te_j)}{4t^2}. \tag{4.69b}$$

These, however, require more function evaluations than their one-sided counterparts (4.68). Nocedal and Wright (2006) suggest that sometimes very little can be gained by employing central rather than one-sided differences.

By incorporating knowledge about the function, especially which components of f depend on which components of x, the number of perturbation vectors and required function evaluations can be reduced. This can yield a significant speed up and should be considered if computational resources are scarce. Further details are explained in Nocedal and Wright (2006).

Even if the partial derivatives on the left hand side of (4.68) and (4.69) do not exist, the finite differences can be computed. The obtained values are then obviously meaningless. There is no feedback to the user that chooses to implement (4.68) or (4.69) without thoroughly investigating $f(x)$, which can be problematic.

4.5.5 Expected Values of Divided Difference Approximations

In the introduction to this section, we have seen that linearization of the function $f(x)$ yields deceivingly simple expressions (4.50) for the expected values of (4.3). If the linearization is achieved by Taylor expansion about $\mathrm{E}\{x\} = \hat{x}$, then the resulting linear model is a local approximation that might be only good in the vicinity of $\hat{x}$. A related discussion is given in Section 4.5.3.

In many applications, including Kalman filtering for nonlinear systems, the available $\hat{x}$ may not be exact. Therefore, Schei (1997) suggested to take also $\operatorname{cov}\{x\} = P$ into account, and to perform a linear interpolation between points that are determined by both $\hat{x}$ and P, rather than only using $\hat{x}$ in a Taylor series.

The expressions of Schei (1997) can also be obtained by replacing the derivatives in the degree-1 Taylor polynomial of $f(x)$ with finite divided differences as in (4.69). The matrix P is taken into account by computing a matrix square root U as in (4.16), with $UU^{\mathrm{T}} = P$, and using the n columns u_i of U as perturbation vectors.

Two papers, also with a focus on Kalman filtering, picked up these ideas and extended them to degree-2 approximations of $f(x)$ that employ divided differences instead of analytical derivatives. Nørgaard et al. (2000b,a) motivates this using a certain interpolation formula. However, both Nørgaard et al. (2000b,a) and Ito and Xiong (2000) sacrifice a number of terms in the approximate solution to the covariance (4.3b) for the sake of a faster algorithm. The suggested solution to (4.3) can be written as

$$\hat{z} \approx \frac{t^2-n}{t^2} f(\hat{x}) + \frac{1}{2t^2}\sum_{i=1}^{n} f(\hat{x}+tu_i) + f(\hat{x}-tu_i), \tag{4.70a}$$

$$\begin{aligned} P_z \approx & \frac{1}{4t^2}\sum_{i=1}^{n}\Big(f(\hat{x}+tu_i) - f(\hat{x}-tu_i)\Big)\Big(\cdot\Big)^{\mathrm{T}} \\ & + \frac{t^2-1}{4t^4}\sum_{i=1}^{n}\Big(f(\hat{x}+tu_i) - 2f(\hat{x}) + f(\hat{x}-tu_i)\Big)\Big(\cdot\Big)^{\mathrm{T}}, \end{aligned} \tag{4.70b}$$

with a scale factor t that determines the interpolation points.

Actually, (4.70a) is of the same functional form as the UT mean in (4.37). We highlight this further in Section 4.6.3. In a way, the divided difference scheme (4.70) stands in between the Taylor polynomial approaches of this section and the deterministic sampling methods of Section 4.4. This is also emphasized in Šimandl and Duník (2009) and Gustafsson and Hendeby (2012).

4.6 A Special Case: Quadratic Function and Gaussian Random Vector

We here investigate the moment computation problem (4.3) for a special case. Namely, we consider $f(x)$ that is composed of degree-2 polynomials in the components of x, and compactly referred to as *quadratic function*. The combination of a quadratic $f(x)$ with a Gaussian random vector x allows for a compact solution to the moment computation problem (4.3), that we employ. However, we also exploit the structure of $f(x)$ to derive a scheme that is fully based on function evaluations, similar to the approaches in Section 4.4 and Section 4.5.5.

4.6.1 Quadratic Functions

We consider $f(x) = [f_1(x), \dots, f_{n_y}(x)]^{\mathrm{T}}$ with scalar components that are given by

$$f_l(x) = a_l + b_l^{\mathrm{T}} x + x^{\mathrm{T}} C_l x. \tag{4.71}$$

Here, the a_l are scalars, the b_l are n-vectors, and the C_l are $n \times n$-matrices, all of them assumed known. Essentially, $f(x)$ is a quadratic polynomial in the components of x, and as such fully represented by its Taylor series. Using $J_l(\overline{x})$ and $H_l(\overline{x})$ as Jacobian and Hessian matrices that contain first and second order partial derivatives, evaluated at $\overline{x}$, we can write

$$f_l(x) = f_l(\overline{x}) + J_l(\overline{x})(x - \overline{x}) + \frac{1}{2}(x - \overline{x})^{\mathrm{T}} H_l(\overline{x})(x - \overline{x}). \tag{4.72}$$

Here, the expansion point $\overline{x}$ is arbitrary. The degree-2 Taylor polynomial is thus not only a local approximation, but exact everywhere.

For a Gaussian $x \sim \mathcal{N}(\hat{x}, P)$, the moments (4.3) are determined by

$$\mathrm{E}\{f_l(x)\} = f_l(\hat{x}) + \frac{1}{2}\,\mathrm{tr}(H_l(\hat{x})P), \tag{4.73a}$$

$$\mathrm{cov}\{f_l(x), f_m(x)\} = J_l(\hat{x})PJ_m(\hat{x})^{\mathrm{T}} + \frac{1}{2}\,\mathrm{tr}(H_l(\hat{x})PH_m(\hat{x})P), \tag{4.73b}$$

with $l = 1, \ldots, n_y$, and $m = 1, \ldots, n_y$. We used $\overline{x} = \hat{x}$ as expansion point, because then odd moments in $(x - \overline{x})$ vanish. The derivations are straightforward but somewhat cumbersome, and the Gaussian assumption is actually only needed for (4.73b). More details are, for instance, given in Bar-Shalom et al. (2001) or Gustafsson and Hendeby (2012).

4.6.2 Sample Implementation of the Expected Values

In this section, we show that (4.73) can be computed exactly by processing a number of deterministic samples in x. In that sense, we establish a connection to UT and schemes that are based on divided differences. The material is based on Gustafsson and Hendeby (2012) and the extension in Roth and Gustafsson (2011). A difference is that we restrict the presentation to quadratic functions, and obtain exact solutions. If $f(x)$ is not quadratic, the presented material yields an approximate computation scheme.

For the following discussion, it suffices to investigate two scalar components of $f(x)$, say the first two rows $f_1(x)$ and $f_2(x)$. Furthermore, we simplify the notation by dropping the arguments of $J_1(\hat{x})$ and $H_1(\hat{x})$. J_1 and H_1, or J_2 and H_2, are by convention evaluated at $\hat{x}$. Furthermore, J denotes the entire Jacobian matrix of $f(x)$, that is given by $J^{\mathrm{T}} = [J_1^{\mathrm{T}}, \ldots, J_{n_y}^{\mathrm{T}}]$.

A key ingredient in the following calculations is, again, a matrix square root U such that $UU^{\mathrm{T}} = P$, as encountered in (4.16). With P being a proper covariance, and as such symmetric and positive definite, U is square with n columns u_i. The matrix P can be expanded as outer product

$$P = \sum_{i=1}^{n} u_i u_i^{\mathrm{T}}. \tag{4.74}$$

Similar to UT, we then select $2n + 1$ deterministic samples in x, or *sigma points*.

Their location is determined by a positive scalar α, that the user selects.

$$x^{(0)} = \hat{x}, \tag{4.75a}$$

$$x^{(\pm i)} = \hat{x} \pm \sqrt{\alpha} u_i, \quad i = 1, \ldots, n. \tag{4.75b}$$

Next, the points (4.75) are passed through f_1. Using the degree-2 Taylor polynomial of f_1 about $\hat{x}$, with help of (4.72), we obtain

$$z_1^{(\pm i)} = f_1(\hat{x}) \pm \sqrt{\alpha} J_1 u_i + \frac{\alpha}{2} u_i^{\mathrm{T}} H_1 u_i. \tag{4.76}$$

Again, we highlight that this is not an approximation but exact for any α, because we consider quadratic functions.

Because of the symmetry of the selected points, we can isolate the Jacobian and Hessian terms by summing transformed sigma points.

$$z^{(i)} - z^{(-i)} = 2\sqrt{\alpha} J u_i, \tag{4.77a}$$

$$z_1^{(i)} + z_1^{(-i)} = 2 f_1(\hat{x}) + \alpha u_i^{\mathrm{T}} H_1 u_i. \tag{4.77b}$$

Here, we dropped the subscript from (4.77a) because it holds for any individual component of z, but also all of them stacked together. After minor rearrangements, and using $z^{(0)} = f(\hat{x})$, we obtain

$$J u_i = \frac{1}{2\sqrt{\alpha}} (z^{(i)} - z^{(-i)}), \tag{4.78a}$$

$$u_i^{\mathrm{T}} H_1 u_i = \frac{1}{\alpha} (z_1^{(i)} - 2 z_1^{(0)} + z_1^{(-i)}). \tag{4.78b}$$

We now proceed to show how (4.78) can be used to compute two terms in (4.73).

Mean value computation: Hessian term

The second term in (4.73a) involves the Hessian matrix. With the outer product representation of P in (4.74), the above established result (4.78a), and the properties of the trace operation, we can calculate

$$\operatorname{tr}(H_1 P) = \sum_{i=1}^{n} \operatorname{tr}(H_1 u_i u_i^{\mathrm{T}}) = \sum_{i=1}^{n} u_i^{\mathrm{T}} H_1 u_i = \frac{1}{\alpha} \sum_{i=1}^{n} (z_1^{(i)} - 2 z_1^{(0)} + z_1^{(-i)}) \tag{4.79}$$

by summing transformed samples.

Covariance computation: Jacobian term

In a similar manner, the outer product representation of P in (4.74) and the isolated Jacobian term in (4.78b) yield

$$J P J^{\mathrm{T}} = \sum_{i=1}^{n} J u_i u_i^{\mathrm{T}} J^{\mathrm{T}} = \frac{1}{4\alpha} \sum_{i=1}^{n} (z^{(i)} - z^{(-i)})(z^{(i)} - z^{(-i)})^{\mathrm{T}}, \tag{4.80}$$

a sum of transformed sigma points for the first term of (4.73b).

Covariance computation: Hessian term

We next address the second term in (4.73b), that involves the Hessian matrices H_1 and H_2 of $f_1(x)$ and $f_2(x)$, respectively. Using (4.74), we can expand it as

$$\operatorname{tr}(H_1 P H_2 P) = \sum_{i=1}^{n} \sum_{j=1}^{n} \operatorname{tr}(H_1 u_i u_i^{\mathrm{T}} H_2 u_j u_j^{\mathrm{T}}) = \sum_{i=1}^{n} \sum_{j=1}^{n} u_j^{\mathrm{T}} H_1 u_i u_i^{\mathrm{T}} H_2 u_j, \tag{4.81}$$

a sum of n^2 terms. One such term is given by

$$u_j^{\mathrm{T}} H_1 u_i u_i^{\mathrm{T}} H_2 u_j. \tag{4.82}$$

For $i = j$, we can reuse the result (4.78b) and obtain

$$u_i^{\mathrm{T}} H_1 u_i u_i^{\mathrm{T}} H_2 u_i = \frac{1}{\alpha^2} (z_1^{(i)} - 2 z_1^{(0)} + z_1^{(-i)})(z_2^{(i)} - 2 z_2^{(0)} + z_2^{(-i)}). \tag{4.83}$$

For the mixed i, j-terms, however, there is no immediate way to write down (4.82) in terms of transformed sigma points. Therefore, we replace the sigma points in (4.75) by an extended set:

$$x^{(0)} = \hat{x}, \tag{4.84a}$$

$$x^{(\pm ij)} = \hat{x} \pm \frac{\sqrt{\alpha}}{2} (u_i + u_j), \quad i = 1, \ldots, n, \quad j = 1, \ldots, n. \tag{4.84b}$$

By construction, the points of (4.75) are contained in (4.84). In addition, $n(n-1)$ unique points are generated. Figure 4.3 illustrates the extended as well as the set of (4.75) for $n = 2$.

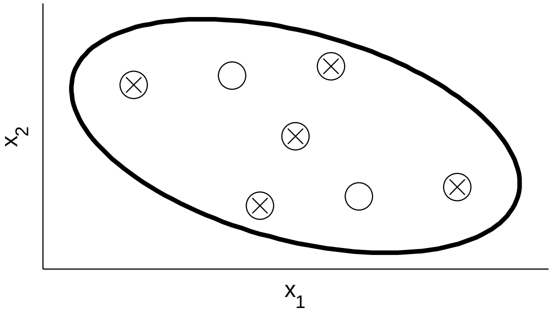


Figure 4.3: *The extended samples of* (4.84) *(circles) and the original samples of* (4.75) *(crosses) for* $n = 2$. *The ellipse indicates a contour for which* $\mathcal{N}(x; \hat{x}, P)$ *is constant. The required matrix square root was calculated using the singular value decomposition of* P.

Similar to (4.76), we obtain an expression for the transformed samples:

$$z_1^{(\pm ij)} = f_1(\hat{x}) \pm \frac{\sqrt{\alpha}}{2} J_1 (u_i + u_j) + \frac{\alpha}{8} (u_i + u_j)^{\mathrm{T}} H_1 (u_i + u_j). \tag{4.85}$$

Again, we can isolate the Hessian term by a combination of transformed samples similar to (4.77b). The resulting expression

$$\begin{aligned} z_1^{(ij)} - 2z_1^{(0)} + z_1^{(-ij)} &= \frac{\alpha}{4}(u_i + u_j)^{\mathrm{T}} H_1 (u_i + u_j) \\ &= \frac{\alpha}{4}(u_i^{\mathrm{T}} H_1 u_i + u_j^{\mathrm{T}} H_1 u_j + 2u_j^{\mathrm{T}} H_1 u_i) \end{aligned} \tag{4.86}$$

involves a mixed term that we require for computing (4.82).

Using that $z^{(i)} = z^{(ii)}$, and (4.78b), we have

$$u_i^{\mathrm{T}} H_1 u_i = \frac{1}{\alpha}(z_1^{(ii)} - 2z_1^{(0)} + z_1^{(-ii)}). \tag{4.87}$$

We can now isolate the mixed term $2u_j^{\mathrm{T}} H_1 u_i$ of (4.86) by subtracting expressions of the form (4.87):

$$\begin{aligned} 2u_j^{\mathrm{T}} H_1 u_i = &\frac{4}{\alpha}(z_1^{(ij)} - 2z_1^{(0)} + z_1^{(-ij)}) - \frac{1}{\alpha}(z_1^{(ii)} - 2z_1^{(0)} + z_1^{(-ii)}) \\ &- \frac{1}{\alpha}(z_1^{(jj)} - 2z_1^{(0)} + z_1^{(-jj)}). \end{aligned} \tag{4.88}$$

The obtained (4.88) merely simplifies to (4.86) for $i = j$. Furthermore, we can simplify (4.88), to obtain a sum of transformed sigma points for

$$u_j^{\mathrm{T}} H_1 u_i = \frac{1}{2\alpha}\left(4z_1^{(ij)} + 4z_1^{(-ij)} - z_1^{(ii)} - z_1^{(-ii)} - z_1^{(jj)} - z_1^{(-jj)} - 4z_1^{(0)}\right). \tag{4.89}$$

Consequently, (4.82) can be written as product

$$\begin{aligned} u_j^{\mathrm{T}} H_1 u_i u_i^{\mathrm{T}} H_2 u_j = &\frac{1}{4\alpha^2}\left(4z_1^{(ij)} + 4z_1^{(-ij)} - z_1^{(ii)} - z_1^{(-ii)} - z_1^{(jj)} - z_1^{(-jj)} - 4z_1^{(0)}\right) \\ &\left(4z_2^{(ij)} + 4z_2^{(-ij)} - z_2^{(ii)} - z_2^{(-ii)} - z_2^{(jj)} - z_2^{(-jj)} - 4z_2^{(0)}\right). \end{aligned} \tag{4.90}$$

The trace term in (4.81) can then be obtained by summation of the above (4.90) over all admissible i and j:

$$\begin{aligned} \mathrm{tr}(H_1 P H_2 P) = &\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{1}{4\alpha^2}\left(4z_1^{(ij)} + 4z_1^{(-ij)} - z_1^{(ii)} - z_1^{(-ii)} - z_1^{(jj)} - z_1^{(-jj)} - 4z_1^{(0)}\right) \\ &\left(4z_2^{(ij)} + 4z_2^{(-ij)} - z_2^{(ii)} - z_2^{(-ii)} - z_2^{(jj)} - z_2^{(-jj)} - 4z_2^{(0)}\right). \end{aligned} \tag{4.91}$$

4.6.3 Summary and Relations to Other Schemes

With regard to the moment computation problem (4.3), and in particular the compact solution for quadratic functions with Gaussian noise (4.73), we now summarize the results of the previous paragraphs.

Using the sigma point formula (4.79) for the trace term in (4.73a), we obtain for the mean

$$\begin{aligned}\hat{z} = \mathrm{E}\{f(x)\} &= z^{(0)} + \frac{1}{2\alpha}\sum_{i=1}^{n}(z^{(i)} - 2z^{(0)} + z^{(-i)}) \\ &= \frac{\alpha - n}{\alpha}z^{(0)} + \frac{1}{2\alpha}\sum_{i=1}^{n}(z^{(i)} + z^{(-i)}). \end{aligned} \tag{4.92a}$$

We have encountered similar expressions earlier in the text. First, the unscented transformation provides a similar solution for $\hat{z}$, that is given in (4.37). Selecting $\kappa = \alpha - n$ in (4.35) gives equivalence between (4.92a) and (4.37). Second, $\hat{z}$ given by the divided difference scheme (4.70a) is equivalent to (4.92a). The only difference is that (4.70a) is formulated in terms of $t = \sqrt{\alpha}$.

We now turn to the calculation of the covariance matrix P_z, which we can formulate in terms of transformed sigma points if we employ the results (4.80) and (4.91). Simply collecting the expressions for all entries in P_z, (4.73b), we obtain

$$\begin{aligned} P_z = \operatorname{cov}\{f(x)\} = \frac{1}{4\alpha}\sum_{i=1}^{n}(z^{(ii)} - z^{(-ii)})(z^{(ii)} - z^{(-ii)})^{\mathrm{T}} + \frac{1}{8\alpha^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \\ \left(4z^{(ij)} + 4z^{(-ij)} - z^{(ii)} - z^{(-ii)} - z^{(jj)} - z^{(-jj)} - 4z^{(0)}\right) \\ \left(4z^{(ij)} + 4z^{(-ij)} - z^{(ii)} - z^{(-ii)} - z^{(jj)} - z^{(-jj)} - 4z^{(0)}\right)^{\mathrm{T}}. \end{aligned} \tag{4.92b}$$

In the derivation of (4.92b), we had to include more sigma points than in UT. As a consequence, the expression for P_z involves summing over more terms. Moreover, if we compare (4.92b) with the divided difference based (4.70b), we can see that many terms have been discarded in the latter. After all, the scheme (4.70) is also based on a quadratic $f(x)$.

Although we focused on a quadratic function in order to derive (4.92), the discussion can be extended to general functions $f(x)$ that are approximated by degree-2 Taylor polynomials. Then, the provided $\hat{z}$ and P_z are, of course, only approximate. Gustafsson and Hendeby (2012) assume this perspective, but still derive some asymptotic relations between UT and calculations that are similar to the above. In Roth and Gustafsson (2011), an analysis of the involved computational complexity is given. There, it is shown that the sigma point calculations can be executed more efficiently than computing (4.73) directly.

4.7 Two Examples

The following two examples put the presented algorithms on test. Whereas the first example considers a mapping $f : \mathbb{R}^2 \to \mathbb{R}^2$, the second example involves a function $f : \mathbb{R}^n \to \mathbb{R}$ for arbitrary n. Both exhibit effects that are far from linear. Furthermore, the input vector x is Gaussian in both examples.

4.7.1 A Quadratic Function

We here consider the moment computation problem (4.3) for quadratic functions with Gaussian input. As described in Section 4.6, there is an analytical solution available, which we can compare to other schemes.

Problem description

We consider a function $f : \mathbb{R}^2 \to \mathbb{R}^2$. Each row of $z = f(x)$ can be written as

$$z_l = f_l(x) = a_l + b_l^{\mathrm{T}} x + x^{\mathrm{T}} C_l x, \quad l = 1, 2. \tag{4.93}$$

As pointed out earlier in Section 4.6, each $f_l(x)$ is fully represented by its degree-2 Taylor polynomial. Furthermore, for a Gaussian $x \sim \mathcal{N}(\hat{x}, P)$, the mean and covariance of $z = f(x)$ are given by (4.73).

The Jacobian vector and Hessian matrix for each component are given by

$$J_l(x) = b_l^{\mathrm{T}} + 2x^{\mathrm{T}} C_l, \tag{4.94a}$$

$$H_l = 2C_l, \quad l = 1, 2. \tag{4.94b}$$

For the following experiments, a_l, b_l, and C_l, are generated randomly. Specifically, the entries in a_l and b_l are drawn from $\mathcal{N}(0, 1)$. The entries in C_i are drawn from $\mathcal{N}(0, 0.01)$ to give a "mild", yet significant nonlinear behavior. Also, the covariance matrix P is generated randomly. In contrast, the mean $\hat{x}$ is zero in all experiments.

Utilized methods

We compare the following methods for computing $\hat{z}$ and P_z:

- MC — The Monte Carlo integration of Section 4.3 with $N = 5000$ samples.
- UT/C3 — The unscented transformation of Section 4.4.1 with $\kappa = 0$, which coincides with the degree-3 integration rule of Section 4.4.2. The weights and samples are given by (4.35).
- UTs — The scaled unscented transformation of Section 4.4.1 with the parameters $\alpha = 1/2$, $\beta = 2$, $\kappa = 0$. The weights and samples are given by (4.38).
- T1 — The expected values of the linearized $f(x)$. Linearization is achieved by using the degree-1 Taylor polynomial. This is explained in Section 4.5.
- T2 — The expected values of the degree-2 Taylor polynomial of $f(x)$. Because $f(x)$ is quadratic, the discussions of Section 4.6 apply.
- D2 — The divided difference based scheme of Section 4.5.5 with $t = 2$.

A priori considerations

With the knowledge that we accumulated throughout this chapter, the following statements can be made before carrying out the actual experiments.

- MC — For large N, we should be able to recover the true $\hat{z}$ and P_z well.

- UT/C3 — The used cubature rule is of degree 3 and as such able to compute $\hat{z}$ correctly. However, computing P_z requires integrating functions of degree 4. Consequently, we can expect errors in P_z.
- UTs — Statements are difficult to make. The performance should depend on the selected parameters as well as $f(x)$.
- T1 — A linearization cannot fully reflect the quadratic function $f(x)$. Therefore, we can expect errors in both $\hat{z}$ and P_z.
- T2 — With $f(x)$ as quadratic function, this method gives the exact $\hat{z}$ and P_z.
- D2 — With the mean value equivalent to T2 for a quadratic function, we can expect the correct $\hat{z}$ but some error in P_z.

Results and discussion

The outcome of two different experiments, with different realizations of $f(x)$ and P, is presented in Figure 4.4 and Figure 4.5. For each experiment, we illustrate the samples by MC, UT/C3, UTs before and after passing them through f. Furthermore, we use the transformed MC samples to obtain a *kernel density estimate* of the true density $p(z)$, which we illustrate as contour plot. For details on this technique, the reader is referred to Bishop (2006). The computed mean vectors $\hat{z}$ and covariance matrices P_z are used to approximate $p(z)$ as Gaussian, and illustrated via 90 percent probability ellipses.

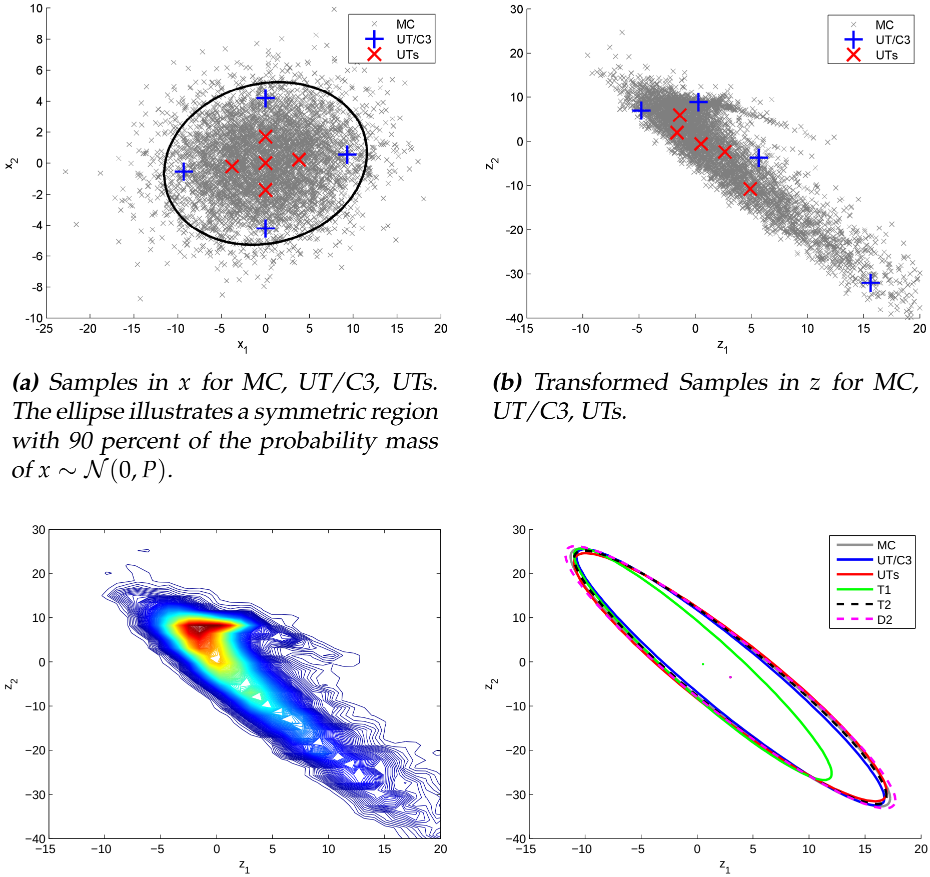
The shown experimental outcomes were chosen to illustrate that the performance of most methods depends on the problem at hand. All methods work well for the first experiment, including T1, as shown in Figure 4.4. Figure 4.5, instead, shows large differences in performance for the second experiment.

We can summarize that all methods but T1 recover the correct $\hat{z}$. A linear approximation is not sufficient here. T2 always provides the correct $\hat{z}$ and P_z because the scheme is specifically tailored to the given f. MC deviates only slightly from the exact solution, due to the random sampling. UT/C3 has a tendency to underestimate P_z in this example. If either C_1 or C_2 are zero, then UT/C3 computes three of four entries in P_z correctly, as we expect from a degree-3 rule. UTs works often better, but is sometimes worse. Although D2 performs well in Figure 4.4 and Figure 4.5, its results are bad in other experiments.

Concluding remarks

This simple (academic) example shows the importance of a thorough analysis of the involved functions, which can be achieved by investigating the Taylor series of $f(x)$. Of course, this only applies if the function allows for such an analysis. After all, $f(x)$ must be available as analytical expression and sufficiently smooth.

We have seen that all of the deterministic schemes, UT/C3, UTs, and D2, fail for some of the experiments. With UT/C3, and the available accuracy statements, this behavior does not come unexpected. The low accuracy of the underlying degree-3 integration rule manifests itself in a bad approximation of P_z.



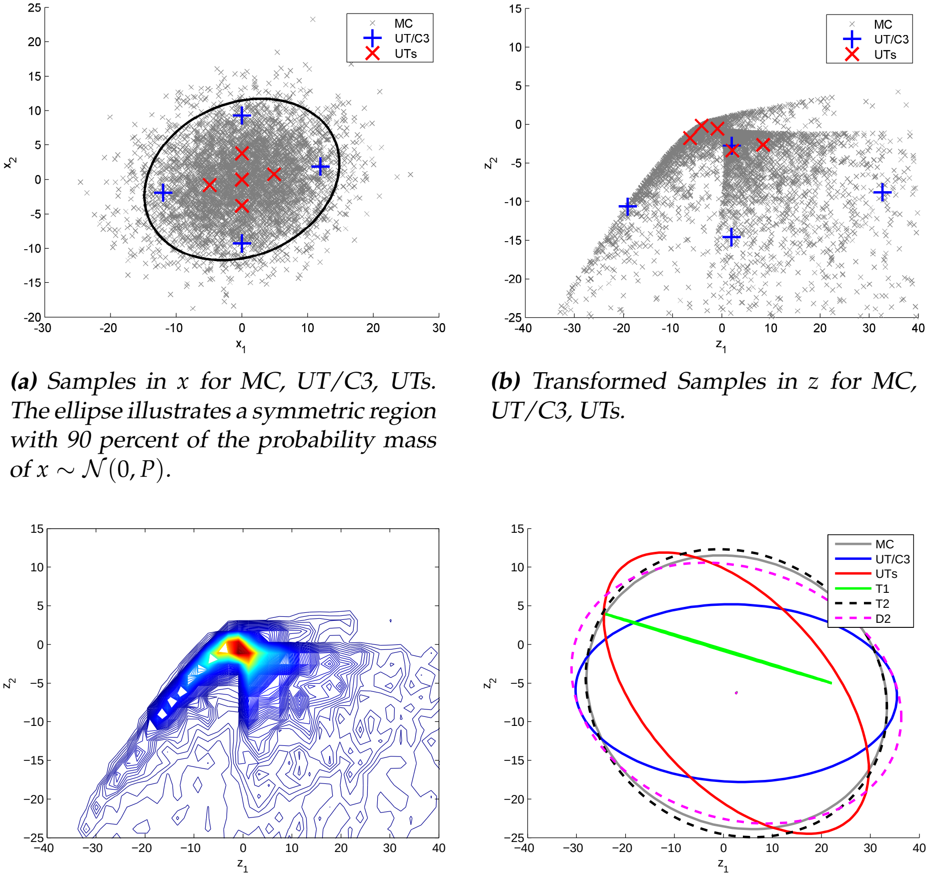
(a) *Samples in x for MC, UT/C3, UTs. The ellipse illustrates a symmetric region with 90 percent of the probability mass of $x \sim \mathcal{N}(0, P)$.*

(b) *Transformed Samples in z for MC, UT/C3, UTs.*

(c) *Kernel density estimate of $p(z)$ based on the transformed MC samples.*

(d) *The ellipses illustrate regions with 90 percent of the probability mass under a Gaussian assumption on $p(z)$.*

Figure 4.4: *The first experiment. The MC samples in (b) show that the mapping is nonlinear. In (d) it can be seen that almost all methods yield similar performance. Even T1 can be seen as reasonable Gaussian approximation of the density in (c). However, T1 has a slightly misplaced $\hat{z}$.*



***(a)** Samples in x for MC, UT/C3, UTs. The ellipse illustrates a symmetric region with 90 percent of the probability mass of $x \sim \mathcal{N}(0, P)$.*

***(b)** Transformed Samples in z for MC, UT/C3, UTs.*

***(c)** Kernel density estimate of $p(z)$ based on the transformed MC samples.*

***(d)** The ellipses illustrate regions with 90 percent of the probability mass under a Gaussian assumption on $p(z)$.*

***Figure 4.5:** The second experiment. The MC samples in (b) and the density in (c) illustrate that the function is more difficult than in the first experiment. In (d), all methods but T1 provide the correct $\hat{z}$. However, the results for P_z differ considerably. T2, being the exact solution, and MC provide ellipses that are very close. T1 totally underestimates the size of P_z, but also UT/C3 and UTs provide a P_z that is too small. D2 is close for this realization of $f(x)$.*

4.7.2 Kullback-Leibler Divergence Minimization

We here continue Example 4.2 for a Gaussian density $p(x)$ and a t density $q(x)$ with given degrees of freedom ν. The problem at hand is that we would like to approximate $p(x)$ by choosing the parameters of $q(x)$. Here, both $p(x)$ and $q(x)$ are elliptically contoured distributions with identical mean value. Therefore, it suffices to look for a positive factor c to adjust the matrix parameter in $q(x)$. The densities are given by

$$p(x) = \mathcal{N}(x; 0, P) = \frac{1}{(2\pi)^{\frac{n}{2}}} \frac{1}{\sqrt{\det(P)}} \exp(-\frac{1}{2} x^{\mathrm{T}} P^{-1} x), \tag{4.95a}$$

$$q(x) = \mathrm{St}(x; 0, cP, \nu) = \frac{\Gamma(\frac{\nu+n}{2})}{\Gamma(\frac{n}{2})} \frac{1}{\sqrt{\det(cP)}} (1 + \frac{1}{\nu} x^{\mathrm{T}} (cP)^{-1} x)^{-\frac{\nu+n}{2}}. \tag{4.95b}$$

Following Example 4.2, we would like to find the factor c by minimization of $\mathrm{KL}(p \| q)$ in (4.7). This corresponds to minimization of

$$-\int p(x) \ln(q(x))\, \mathrm{d}x = -\int \mathcal{N}(x; 0, P) \underbrace{\ln(\mathrm{St}(x; 0, cP, \nu))}_{f(x)} \mathrm{d}x = J(c), \tag{4.96}$$

with respect to c. Obviously, (4.96) involves an expected value with respect to a Gaussian density. If we can efficiently evaluate it, a numerical search for c is possible. We will show that this indeed makes sense by investigating the cost function $J(c)$ at the end of this section.

The above problem is of interest to users who want to incorporate the t distribution without deviating too much from Gaussian behavior. One example is given by replacing Gaussian noise in particle filter applications. With the Gaussian distribution being a special case of the t distribution, the example is also one instance of the more general problem of fitting one t density by another t density, that is encountered in the filter development of Chapter 5.

Although the underlying minimization problem is the core motivation for this example, we henceforth focus on the required expected value of $f(x)$ in (4.96). The illustrated results are obtained with with $c = 1$ and $\nu = 4$, randomly generated $n \times n$ covariance matrices P, and $n = 5$ as well as $n = 15$.

Structure of the problem

We first note that the density $p(x) = \mathcal{N}(x; 0, P)$ is elliptically contoured. Consequently, the results of Section 4.2 apply.

In fact, the considered $f(x) = g(x^{\mathrm{T}} P^{-1} x)$ depends only on x via $x^{\mathrm{T}} P^{-1} x$. This facilitates re-writing the n-dimensional expected value into a scalar integral over the range component. The procedure is outlined in Section 4.2 and carried out as follows:

$$
\begin{aligned}
&\int \mathcal{N}(x;0,P) f(x)\,\mathrm{d}x \\
&= \int \underbrace{\mathcal{N}(y;0,I_n)}_{h(y^\mathrm{T}y)} \overbrace{\ln\left(\frac{\Gamma(\frac{\nu+n}{2})}{\Gamma(\frac{n}{2})}\frac{1}{\sqrt{\det(cP)}}(1+\frac{1}{c\nu}y^\mathrm{T}y)^{-\frac{\nu+n}{2}}\right)}^{f(Uy)=g(y^\mathrm{T}y)=g(x^\mathrm{T}P^{-1}x)}\mathrm{d}y \qquad (4.97a) \\
&= \underbrace{\frac{2\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2})}}_{C(n)} \int_0^\infty \underbrace{\frac{1}{(2\pi)^{\frac{n}{2}}}\exp(-\frac{r^2}{2})}_{h(r^2)} g(r^2) r^{n-1}\,\mathrm{d}r. \qquad (4.97b)
\end{aligned}
$$

In (4.97a), stochastic decoupling is applied, and (4.97b) is a result of a change to spherical coordinates and subsequently integrating out all angles.

Taking the expressions as they are, we can employ a computer algebra system such as Mathematica to find an analytical solution for a given P. Although no solution is found for direct integration of (4.96), Mathematica solves (4.97a) for low dimensions, say $n = 5$. As for the scalar integration (4.97b), an exact solution is provided by Mathematica in no time.

Another important consequence of the problem structure is that we can employ scalar Monte Carlo integration by working on $g(x^\mathrm{T}P^{-1}x)$ directly. The random variable $x^\mathrm{T}P^{-1}x$ admits a chi-squared distribution with n degrees of freedom $\chi^2(n)$, from which random samples are easily drawn. When compared to direct Monte Carlo integration of $f(x)$, working on $g(x^\mathrm{T}P^{-1}x)$ requires less computations at comparable accuracy.

Utilized methods

The following methods are employed to compute the expected value of $f(x)$:

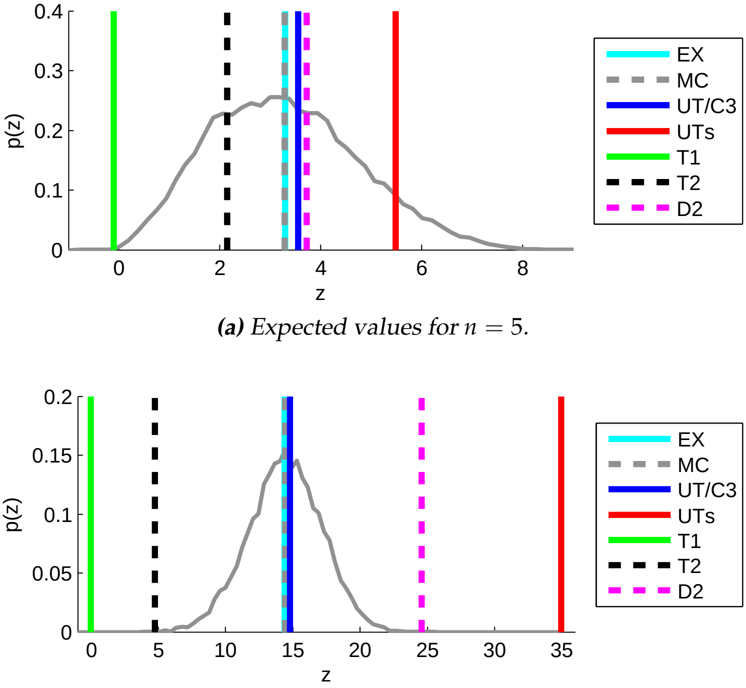
- EX — Exact solution of the expected value in Mathematica. Here, the reduction to a scalar problem (4.97b) facilitates the exact computation for large n.
- MC — Monte Carlo integration as presented in Section 4.3. We present the results of MC integration of $f(x)$ with $N = 10000$ random samples in x.
- UT/C3 — The "standard" UT of Section 4.4.1 with selection scheme (4.35) and $\kappa = 0$. This coincides with the degree-3 integration rule of Section 4.4.2.
- UTs — The scaled unscented transformation of Section 4.4.1 with selection scheme (4.38) and $\alpha = 0.1$, $\beta = 2$, $\kappa = 0$.
- T1 — The expected values of the linearized $f(x)$. Linearization is achieved by using the degree-1 Taylor polynomial. This is explained in Section 4.5. The required derivatives are approximated by central divided differences, as discussed in Section 4.5.4.
- T2 — The expected values of the degree-2 Taylor polynomial of $f(x)$. We

employ the expression (4.73) of Section 4.6. Similar to T1, the required derivatives are approximated by central divided differences.

- D2 — The divided difference based scheme of Section 4.5.5 with $t = 2$.

Results and discussion

Similar to the example of Section 4.7.1, we have an exact method that provides the true expected value, namely the Mathematica based EX. The results for two experiments with $n = 5$ and $n = 15$, and randomly selected matrices P, are given in Figure 4.6. The illustrated $p(z)$ is a *kernel density estimate* and computed based on the MC samples in z. Further details can be found in Bishop (2006). The computed mean values by each scheme are illustrated by vertical lines. Basically, all deterministic methods but UT/C3 fail to give a decent mean estimate. MC integration performs best but also has the highest computational expenses.



(a) *Expected values for $n = 5$.*

(b) *Expected values for $n = 15$.*

Figure 4.6: *Expected values for different n for each of the methods. The MC samples are furthermore used to compute an estimate of the density $p(z)$. The solutions of EX and MC are close in (a) and (b). Also UT/C3 appears close in both cases. Whereas D2 appears still close in (a), all approximate methods but MC and UT/C3 fail to provide accurate results. In both (a) and (b) we selected $c = 1$ and $\nu = 4$. The matrices P are generated randomly.*

The fact that the numerical minimization of (4.96) makes sense is indicated in

Figure 4.7, where we display (4.96) as function of c. The required expected values are computed using MC integration. Indeed, the illustrated $J(c)$ appears convex with a unique minimum. Interestingly, the minimizing c does not seem to depend on the utilized P but merely on n and c. This is especially interesting for online applications, because c can be computed offline and stored for given n and ν, and will turn out useful in Chapter 5.

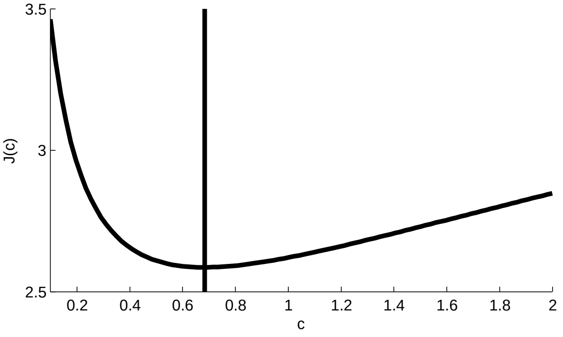


Figure 4.7: *The cost function $J(c)$ of (4.96) for $n = 2$ and $\nu = 3$. The vertical line illustrates the minimizing c.*

4.8 Conclusions

We provided a number of tricks and methods that attempt to solve the moment computation problem (4.3).

- The variable changes in Section 4.2 can be used to restructure problems that involve elliptically contoured distributions. In some cases, it is possible to turn an n-dimensional expected value into a scalar problem that may even admit an analytical solution.

- The Monte Carlo methods of Section 4.3 can be applied to any distribution as long as it can be sampled from. With large computational resources, they are appealing because of their asymptotic properties for large sample ensembles, given by the law of large numbers. If an n-dimensional problem can be broken down to a scalar problem, as mentioned above, the computational and storage demands of MC methods can be decreased immensely.

- Although the unscented transform of Section 4.4 is appealing because of its simple implementation, the lack of accuracy statements is often considered as drawback. The presented deterministic integration rules of the same section, instead, come with concise accuracy statements for polynomial functions. The popular degree-3 rule that we present has still very limited accuracy when compared to Monte Carlo methods, but scales well with the dimension n of x.

- The Taylor series approach of Section 4.5 can be used to gather insight into given problems by comparing the involved function with its degree-d Taylor polynomial in an area of interest. Furthermore, expected value computations of arbitrary accuracy are facilitated by the compact expressions in Section 4.5.

- The scheme and insights in Section 4.6 establish a number of connections between several of the considered methods.
- The examples of Section 4.7 show that there is not one single deterministic moment computation scheme that outperforms all others. Therefore, all algorithms should be applied with extra care. Helpful is also a detailed analysis of the involved functions $f(x)$, and solid knowledge of the accuracy a method can provide. That is, if statements are available.

A main motivation for considering the moment computation problem (4.3) is given by the required expected values in the nonlinear Kalman filters of Section 3.2. We close this chapter with the cautionary remark that computing the required moments is important in such a filter, but that computing accurate moments might still yield an algorithm that does not work. For example, the underlying filtering problem might not be solvable by a Kalman filter in the first place, because the true underlying posterior is multi-modal.

5

Filtering in Heavy-Tailed Noise Using the *t* Distribution

This section describes the development of a filtering algorithm for linear state space models with heavy-tailed process and measurement noise. Our work is based on Student's t distribution, which is described in detail in Appendix A.3. The resulting algorithm turns out to be a generalization of the ubiquitous Kalman filter, but with observation dependent matrix parameters in contrast to the covariance in the Kalman filter. Parts of the material have been published in Roth et al. (2013).

5.1 Motivation and Background

Among signal processing researchers, the Gaussian distribution is not only popular due to its many convenient results and closed form expressions, but also because of the fact that many small arbitrarily distributed noise effects add up to one Gaussian random variable. The latter result is known as the *central limit theorem* and in many cases justifies a Gaussian assumption on certain signals.

There are, however, phenomena that cannot be modeled with a Gaussian distribution. *Outliers*, by which we mean values far from the mean value that might occur occasionally, cannot be generated by a Gaussian distribution at such a rate. Because such outliers commonly appear in many signal processing applications, it is reasonable to investigate distributions with heavier tails than the Gaussian. The t distribution of Appendix A.3 can exhibit heavy tails, but still shares a number of attractive properties with the Gaussian. After all, the Gaussian distribution can be obtained as limiting case of the t distribution with infinite degrees of freedom.

In the context of filtering in state space models, an algorithm that allows for heavy-tailed process noise and heavy-tailed measurement noise is of substantial practical value. This is because many real-world nuisance effects can be modeled by noise outliers, including clutter measurements from unreliable sensors, abrupt changes of unknown input signals, and effects that are caused by simplified modeling of complex real world processes. Often, the prospect of a heavy-tailed state distribution seems preferable over a Gaussian, especially in approximate filters for uncertain or nonlinear models.

We investigate filtering in linear systems, a field in which the Kalman filter of Section 3.1 is optimal in the sense of being the best unbiased linear filter for arbitrary noise with known statistics. Any filter that yields an improvement must therefore be nonlinear. Furthermore, a serious competitor to the Kalman filter should be as simple to implement and as scalable to high-dimensional problems. The particular choice of the *t* distribution, in conjunction with minor approximate adjustments, does result in a nonlinear filter. Also, heavy-tailed noise is addressed in a natural manner. Interestingly, the developed filter reduces to the Kalman filter for one choice of parameters. This is in line with the result that the *t* distribution converges to the Gaussian distribution for infinite degrees of freedom. The computational cost compares to that of the Kalman filter.

Earlier efforts to derive filters based on the *t* distribution were made in Meinhold and Singpurwalla (1989) and Girón and Rojano (1994), but did not receive much attention in the signal processing community. Our approach differs considerably from them, because we include careful approximation steps to obtain closed form expressions. Several recent publications address the filtering problem with heavy-tailed measurement noise only. Whereas Agamennoni et al. (2012) and Piché et al. (2012) employ variational methods to estimate the unknown measurement noise statistics, an optimization approach using ℓ_1-regularization is sketched in Mattingley and Boyd (2010). Common to them is the assumption of well-behaved process noise. This is easily violated in realistic scenarios, for instance tracking of maneuvering targets, and leaves the suggested methods debatable for certain applications.

5.2 Filter Development

In this section, we derive a filter that is fully based on the *t* distribution. Both the measurement noise and the process noise are heavy-tailed. The filtering density $p(x_k|y_{1:k})$ is a *t* density at each time k, and in that respect the algorithm can be viewed as an *assumed density filter*. First, we present an exact solution under somewhat restrictive modeling assumptions. In a second stage, the algorithm is adjusted to cope with more general scenarios.

We consider linear state space models (2.3), here restated for convenience:

$$x_{k+1} = F_k x_k + v_k, \tag{5.1a}$$
$$y_k = H_k x_k + e_k. \tag{5.1b}$$

The state is an n_x-vector, the measurement an n_y-vector, the noise signals v_k and e_k have appropriate dimensions. The matrices F_k and H_k are known throughout time. If the model is nonlinear, F_k and H_k might be obtained by linearization.

The initial state and the noise signals are mutually uncorrelated and t distributed with marginal densities

$$p(x_0) = \text{St}(x_0; \hat{x}_0, P_0, \eta_0), \tag{5.2a}$$

$$p(v_k) = \text{St}(v_k; 0, Q_k, \gamma_k), \tag{5.2b}$$

$$p(e_k) = \text{St}(e_k; 0, R_k, \delta_k). \tag{5.2c}$$

The degrees of freedom η_0, γ_k, δ_k determine the tail behavior of the related densities. For example, a large η_0 yields $p(x_0)$ that is almost Gaussian, and $\eta_0 = 3$ gives heavy-tails but still assures that x_0 has a finite covariance.

5.2.1 An Exact Filter for a Special Case

Similar to the Kalman filter of Section 3.1, we divide the filtering recursions into a time and a measurement update. The parameters of the marginal densities (5.2) are at each time k such that we can apply the results for the t distribution, listed in Appendix A.3. We make use of the rules for affine transformations (A.27), marginalization (A.29), and conditioning (A.30).

Time update

If the degrees of freedom for the process noise $\gamma_k = \eta_k$ for all k, with η_k as degrees of freedom of $x_k|y_{1:k}$, we can formulate a joint density

$$p(x_k, v_k | y_{1:k}) = \text{St}(\begin{bmatrix} x_k \\ v_k \end{bmatrix}; \begin{bmatrix} \hat{x}_{k|k} \\ 0 \end{bmatrix}, \begin{bmatrix} P_{k|k} & 0 \\ 0 & Q_k \end{bmatrix}, \eta_k). \tag{5.3}$$

Here, the state x_k and v_k are uncorrelated but not independent. This is in accordance with the assumptions that are used to derive the Kalman filter from a least-squares perspective, as in Kailath et al. (2000).

The parameters of the prediction density

$$p(x_{k+1}|y_{1:k}) = \text{St}(x_{k+1}; \hat{x}_{k+1|k}, P_{k+1|k}, \eta_k) \tag{5.4}$$

can be computed by a linear transformation and subsequent marginalization:

$$\hat{x}_{k+1|k} = F_k \hat{x}_{k|k}, \tag{5.5a}$$

$$P_{k+1|k} = F_k P_{k|k} F_k^{\mathrm{T}} + Q_k. \tag{5.5b}$$

The parameter η_k is retained. The obtained (5.5) is similar to the Kalman filter time update (3.2).

Measurement update

Again, we make assumptions on the degrees of freedom in (5.2). For $\delta_k = \eta_{k-1}$, we can formulate a joint density for the predicted state and the measurement noise

$$p(x_k, e_k|y_{1:k-1}) = \mathrm{St}(\begin{bmatrix} x_k \\ e_k \end{bmatrix}; \begin{bmatrix} \hat{x}_{k|k-1} \\ 0 \end{bmatrix}, \begin{bmatrix} P_{k|k-1} & 0 \\ 0 & R_k \end{bmatrix}, \eta_{k-1}). \tag{5.6}$$

Then, a linear transformation gives the joint density of the state and the measurement

$$p(x_k, y_k|y_{1:k-1}) = \mathrm{St}(\begin{bmatrix} x_k \\ y_k \end{bmatrix}; \begin{bmatrix} \hat{x}_{k|k-1} \\ H_k\hat{x}_{k|k-1} \end{bmatrix}, \begin{bmatrix} P_{k|k-1} & P_{k|k-1}H_k^{\mathrm{T}} \\ H_kP_{k|k-1} & S_k \end{bmatrix}, \eta_{k-1}), \tag{5.7}$$

with $S_k = H_kP_{k|k-1}H_k^{\mathrm{T}} + R_k$.

The conditional density of the state given all measurements $y_{1:k}$ can be obtained using (A.30) and is again t:

$$p(x_k|y_k, y_{1:k-1}) = \mathrm{St}(x_k; \hat{x}_{k|k}, P_{k|k}, \eta_k). \tag{5.8}$$

The updated parameters are given by

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + P_{k|k-1}H_k^{\mathrm{T}}S_k^{-1}(y_k - H_k\hat{x}_{k|k-1}), \tag{5.9a}$$

$$\Delta_{y,k}^2 = (y_k - H_k\hat{x}_{k|k-1})^{\mathrm{T}}S_k^{-1}(y_k - H_k\hat{x}_{k|k-1}), \tag{5.9b}$$

$$P_{k|k} = \frac{\eta_{k-1} + \Delta_{y,k}^2}{\eta_{k-1} + n_y}(P_{k|k-1} - P_{k|k-1}H_k^{\mathrm{T}}S_k^{-1}H_kP_{k|k-1}), \tag{5.9c}$$

$$\eta_k = \eta_{k-1} + n_y. \tag{5.9d}$$

The similarities to the Kalman filter are apparent. In particular, the linear update of the state estimate (5.9a) and the matrix in (5.9c) can be related to the Kalman filter measurement update (3.3). However, (5.9c) depends nonlinearly on the observation y_k.

Because of the compatible noise properties at each time k, we were able to formulate (5.3) and (5.6), and obtained closed form expressions for the posterior density $p(x_k|y_{1:k})$. The algorithm that is composed of (5.5) and (5.9), initialized with the parameters of (5.2a), thus solves the Bayesian filtering recursions. The estimate $\hat{x}_{k|k}$ of (5.9a) unifies the mean, the mode, and the median of the true posterior. Following Section 2.4, it is therefore optimal in several ways. The filter involves the squared residual (5.9b) and is hence nonlinear in y_k.

With each measurement update (5.9), the degrees of freedom increase according to (5.9d). That, in turn, requires the noise degrees of freedom in (5.2) to increase as well. The problem becomes more and more Gaussian. Indeed, the algorithm converges to the Kalman filter after few time steps.

Analysis of the matrix parameter

The matrix update (5.9c) is a scaled version of the covariance in the Kalman filter (3.3d). For the sake of a simpler expression, we drop all subscripts for the moment.

The scale factor is given by

$$\frac{\eta + \Delta^2}{\eta + n} = \frac{\eta}{\eta + n} + \frac{n}{\eta + n}\frac{\Delta^2}{n}. \tag{5.10}$$

In the limit for $\eta \to \infty$, (5.10) is one, and (5.9) is the Kalman filter measurement update.

With $y_k \sim \mathrm{St}(y_k; H_k \hat{x}_{k|k-1}, S_k, \eta_{k-1})$, we know from Section 4.2 or Appendix A.3 that $\frac{\Delta^2}{n}$ admits the F distribution $\mathcal{F}(n, \eta)$ of Section A.2. Hence, we can compute the mean value

$$\mathrm{E}\left\{\frac{\Delta^2}{n}\right\} = \frac{\eta}{\eta - 2}, \quad \eta > 2, \tag{5.11}$$

and insert it into (5.10), to obtain

$$\mathrm{E}\left\{\frac{\eta + \Delta^2}{\eta + n}\right\} = \frac{\eta}{\eta + n} + \frac{n}{\eta + n}\frac{\eta}{\eta - 2} > 1, \quad \eta > 2. \tag{5.12}$$

With the factor larger than one in the mean, we have shown that the filter is not a "Kalman filter on average". More insights could be gained by deriving the distribution of (5.10) using the transformation theorem of Appendix A.1.4.

5.2.2 The Suggested Heavy-Tailed Filter

The requested conditions on (5.2) in Section 5.2.1 are hardly ever met in practice. We therefore introduce careful approximations that yield a filter algorithm that is only slightly more involved than the exact filter in (5.5) and (5.9). Also, we prevent the degrees of freedom from increasing and thereby keep a heavy-tailed posterior density $p(x_k|y_{1:k})$ throughout time.

Problems with incompatible degrees of freedom

We consider again the linear model (5.1), but this time under the more realistic assumption that the degrees of freedom γ_k and δ_k in (5.2) are arbitrary. Unfortunately, this prohibits closed form time and measurement updates. We present the conflict for a time update.

Suppose that we have the following posterior density for time k:

$$p(x_k|y_{1:k}) = \mathrm{St}(x_k; \hat{x}_{k|k}, P_{k|k}, \eta_k). \tag{5.13}$$

In the derivation of the exact filter in Section 5.2.1, we formulated the joint density of x_k and v_k, given $y_{1:k}$. If $\eta_k \neq \gamma_k$, then $p(x_k, v_k|y_{1:k})$ cannot be expressed in closed form unless v_k is independent. For independent noise, however, the joint density

$$p(x_k, v_k|y_{1:k}) = \mathrm{St}(x_k; \hat{x}_{k|k}, P_{k|k}, \eta_k)\,\mathrm{St}(v_k; 0, Q_k, \gamma_k) \tag{5.14}$$

is no longer a t density, and actually not even elliptically contoured. Our efforts to derive compact time update equations are thus shattered.

Two simplifying approximation steps

What led to the convenient expressions in the exact filter, (5.5) and (5.9), were the joint *t* densities in (5.3) and (5.6). Therefore, it is reasonable to force the actual state and noise densities into this format.

For the time update, we suggest to find a common degrees of freedom parameter $\widetilde{\eta}_k$ from η_k and γ_k. Similar to (5.3), we then use

$$p(x_k, v_k | y_{1:k}) \approx \mathrm{St}(\begin{bmatrix} x_k \\ v_k \end{bmatrix}; \begin{bmatrix} \hat{x}_{k|k} \\ 0 \end{bmatrix}, \begin{bmatrix} \widetilde{P}_{k|k} & 0 \\ 0 & \widetilde{Q}_k \end{bmatrix}, \widetilde{\eta}_k), \tag{5.15}$$

a *t* density approximation. In order to account for the changed degrees of freedom, the matrices $P_{k|k}$ and Q_k are replaced by $\widetilde{P}_{k|k}$ and $\widetilde{Q}_k$. The mean values remain unchanged. Methods to find $\widetilde{P}_{k|k}$ and $\widetilde{Q}_k$ are explained in Section 5.2.3.

The merits of the approximation are substantial, because we can now apply a time update similar to (5.5).

For the measurement update, a similar approximation is suggested:

$$p(x_k, e_k | y_{1:k-1}) \approx \mathrm{St}(\begin{bmatrix} x_k \\ e_k \end{bmatrix}; \begin{bmatrix} \hat{x}_{k|k-1} \\ 0 \end{bmatrix}, \begin{bmatrix} \widetilde{P}_{k|k-1} & 0 \\ 0 & \widetilde{R}_k \end{bmatrix}, \overline{\eta}_{k-1}). \tag{5.16}$$

Given (5.16), the steps to obtain update expressions similar to (5.9) are the same as in the exact filter.

The two density approximations (5.15) and (5.16) extend the exact filter of Section 5.2.1, and provide convenient closed form solutions for the time and the measurement update. Because the measurement update (5.9) is based on the assumption that the predicted state and the predicted measurement are jointly *t* distributed, the filter can be seen as an *assumed density filter*. The term is often used for Kalman filters for nonlinear models — these usually assume Gaussian densities, see Section 3.2 — but appears appropriate here as well. The approximations are mild in the sense that the approximated marginal densities remain *t* densities, yet with possibly altered degrees of freedom.

Algorithm summary

For the sake of a compact presentation, the time and the measurement update are summarized below.

The time update relies on the parameters of a previous posterior density $\hat{x}_{k|k}$, $P_{k|k}$, and η_k. The following line

$$P_{k|k}, Q_k, \eta_k, \gamma_k \to \widetilde{P}_{k|k}, \widetilde{Q}_k, \widetilde{\eta}_k \tag{5.17a}$$

denotes the systematic choice of the approximated from the original parameters.

The update of the mean and matrix parameter is given by

$$\hat{x}_{k+1|k} = F_k \hat{x}_{k|k}, \tag{5.17b}$$

$$P_{k+1|k} = F_k \widetilde{P}_{k|k} F_k^{\mathrm{T}} + \widetilde{Q}_k. \tag{5.17c}$$

The measurement update relies on previously obtained parameters $\hat{x}_{k|k-1}$, $P_{k|k-1}$, and $\widetilde{\eta}_{k-1}$. If required, an approximation step

$$P_{k|k-1}, R_k, \widetilde{\eta}_{k-1}, \delta_k \rightarrow \widetilde{P}_{k|k-1}, \widetilde{R}_k, \overline{\eta}_{k-1} \tag{5.18a}$$

is applied. Then, the posterior parameters are computed according to

$$S_k = H_k \widetilde{P}_{k|k-1} H_k^{\mathrm{T}} + \widetilde{R}_k, \tag{5.18b}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \widetilde{P}_{k|k-1} H_k^{\mathrm{T}} S_k^{-1} (y_k - H_k \hat{x}_{k|k-1}), \tag{5.18c}$$

$$\overline{P}_{k|k} = \widetilde{P}_{k|k-1} - \widetilde{P}_{k|k-1} H_k^{\mathrm{T}} S_k^{-1} H_k \widetilde{P}_{k|k-1}, \tag{5.18d}$$

$$\Delta_{y,k}^2 = (y_k - H_k \hat{x}_{k|k-1})^{\mathrm{T}} S_k^{-1} (y_k - H_k \hat{x}_{k|k-1}), \tag{5.18e}$$

$$P_{k|k} = \frac{\overline{\eta}_{k-1} + \Delta_{y,k}^2}{\overline{\eta}_{k-1} + n_y} \overline{P}_{k|k}, \tag{5.18f}$$

$$\eta_k = \overline{\eta}_{k-1} + n_y. \tag{5.18g}$$

The presentation in (5.18) highlights the relations to the Kalman filter measurement update (3.3), which is in fact contained in (5.18b–5.18d).

For the user, it is convenient to set the degrees of freedom parameters in (5.2) to a constant value that reflects the desired noise properties for all times. That is: $\gamma_k = \delta_k = \eta_0 = \nu$ for all k. As a consequence, with $\widetilde{\eta}_k = \nu$ in (5.17a), the degrees of freedom are compatible in the measurement update and (5.18a) can be skipped. The remaining approximation step can be interpreted as forcing the degrees of freedom down to one assumed value ν after the measurement update. This is what keeps the algorithm from converging to the Kalman filter.

5.2.3 Required Approximation Steps

Both approximation steps (5.17a) and (5.18a) require the choice of a common degrees of freedom parameter, and an adjustment of matrix parameters. We suggest to first chose the degrees of freedom and then adjust each marginal density separately. For (5.17a) that amounts to first choosing $\widetilde{\eta}_k$, and then computing $\widetilde{P}_{k|k}$ and $\widetilde{Q}_k$ independently.

There are certainly more sophisticated schemes to find the common degrees of freedom in (5.17a) and (5.18a), but a simple scheme that preserves the heaviest tails of the individual marginal densities is given by simply taking the minimum of both degrees of freedom parameters. For (5.17a) that amounts to selecting $\widetilde{\eta}_k = \min(\eta_k, \gamma_k)$.

With regard to the problem of adjusting the marginal matrix parameters given new degrees of freedom, for example finding $\widetilde{P}_{k|k}$ given $\widetilde{\eta}_k$ in (5.17a), we can argue that qualitative characteristics should be preserved. Consequently, the adjusted matrix parameter should be a scaled version of the original matrix. Formulated as generic problem, we must find a scalar $c > 0$ such that the densities in

$$p(z) = \mathrm{St}(z; 0, P, \nu), \tag{5.19a}$$

$$q(z) = \mathrm{St}(z; 0, cP, \widetilde{\nu}), \tag{5.19b}$$

are close in some to be specified sense. Here, it is desirable to have a closeness criterion that involves as few user choices as possible. We present two approaches that take $p(z)$ and $q(z)$ as they are without involving any user interaction.

Approximation by Kullback-Leibler divergence minimization

We have already encountered such density fitting problems in Chapter 4, first briefly outlined in Section 4.1 and later as worked out example in Section 4.7.2. The there suggested approach is minimization of the *Kullback-Leibler divergence* between $p(z)$ and $q(z)$. This can be achieved if the expected value

$$\int p(x) \ln\left(q(x)\right) \, \mathrm{d}x \tag{5.20}$$

can be efficiently computed, because then a numerical search for c is possible.

In Section 4.7.2, $p(x)$ is Gaussian and (5.20) can be accurately computed using Monte Carlo integration. This is achieved by either working on the n-vector x directly, or by utilizing that $q(x)$ merely depends on x via the chi-squared random variable $x^{\mathrm{T}} P^{-1} x$ for Gaussian $p(x)$. These results are easily extended to a t density $p(x)$ in (5.19a). Monte Carlo integration using x directly is achieved by processing n-dimensional t random variables, and the more efficient scalar scheme utilizes the fact that $x^{\mathrm{T}} P^{-1} x / n$ is an F random variable.

Hence, we can employ a numerical search to determine c. Luckily, the empirical result of Section 4.7.2 that c depends only on the involved degrees of freedom and the dimension n, carries over. The minimizing c depends only on n, ν, and $\tilde{\nu}$.

The approximations in a filtering scenario

For a filter algorithm that utilizes constant degrees of freedom throughout time, the scale factors can be computed offline. If $\gamma_k = \delta_k = \eta_0 = \nu$, then we need only approximate $P_{k|k}$ after each measurement update to obtain $\widetilde{P}_{k|k} = cP_{k|k}$. If we find c by minimizing the Kullback-Leibler divergence between the densities

$$p(z) = \mathrm{St}(z; 0, P_{k|k}, \nu + n_y), \tag{5.21a}$$

$$q(z) = \mathrm{St}(z; 0, cP_{k|k}, \nu), \tag{5.21b}$$

then that c can be computed offline and used throughout time.

Approximation by moment matching

A simple and popular alternative to finding c by minimization of the Kullback-Leibler divergence is matching of the covariance matrices that are related to $p(z)$

and $q(z)$. Often, this is referred to as *moment matching*. Again, no user interaction is required. From the covariance of t random variables (A.25), we obtain the condition

$$\frac{\nu}{\nu-2}P = \frac{\widetilde{\nu}}{\widetilde{\nu}-2}cP, \tag{5.22}$$

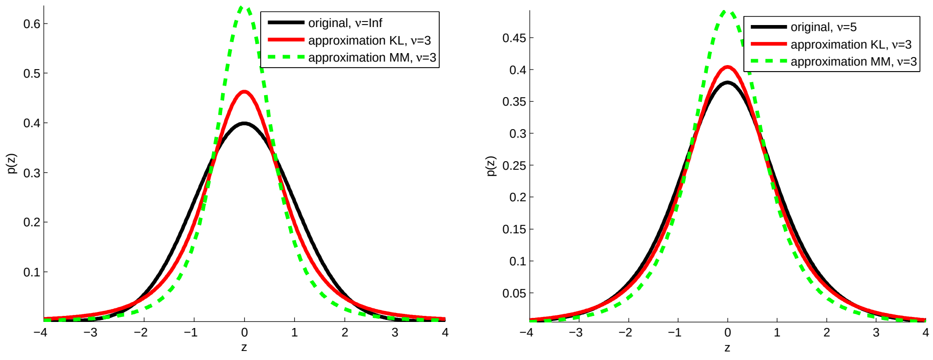
for $\nu > 2$ and $\widetilde{\nu} > 2$. The resulting scale factor c is then given by

$$c = \frac{\nu(\widetilde{\nu}-2)}{(\nu-2)\widetilde{\nu}}. \tag{5.23}$$

The problem with moment matching is that the covariance matrices are heavily influenced by the tail behavior of the involved distributions. If ν is large and $\widetilde{\nu}$ is small, then the $q(z)$ with matched covariance is considerably more peaked around its mean than $p(z)$. The following example shows why the Kullback-Leibler divergence scheme is often preferable over covariance matching.

Example 5.1: Comparison of approximation methods

In this example, we investigate one-dimensional fitting problems. First, we consider a Gaussian density $p(z) = \mathcal{N}(z; 0, 1)$, which we want to approximate by a t density with 3 degrees of freedom $q(x) = \mathrm{St}(z; 0, c, 3)$. The scale factor c is found by two methods: KL denotes minimization of the Kullback-Leibler divergence between $p(z)$ and $q(z)$ with respect to c, and MM denotes matching the variances of $p(z)$ and $q(z)$. Second, we repeat the experiment with $p(z) = \mathrm{St}(z; 0, 1, 5)$. The results are illustrated in Figure 5.1a.



(a) $\mathcal{N}(z; 0, 1) = \lim_{\nu\to\infty} \mathrm{St}(z; 0, 1, \nu)$ *and two approximating densities* $\mathrm{St}(z; 0, c, 3)$.

(b) $\mathrm{St}(z; 0, 1, 5)$ *and two approximating densities* $\mathrm{St}(z; 0, c, 3)$.

Figure 5.1: *In (a), a Gaussian density (black, solid) is illustrated along two* t *densities. The* t *densities have 3 degrees of freedom and are obtained by minimization of the Kullback-Leibler divergence between* t *and Gaussian (red, solid), and matching the variance of* t *and Gaussian (green, dashed). In (b), the Gaussian density is replaced by a* t *density with 5 degrees of freedom.*

In both Figure 5.1a and Figure 5.1b, the MM approximation appears more peaked around the mean. KL seems to achieve a better trade-off between the approximation in the tails and close to the mean.

5.3 An Example Application: Tracking Maneuvering Targets in Clutter

In this section, we test the *t* filter that we developed on a challenging example and compare its performance to state-of-the-art algorithms. The problem is taken from the field of target tracking and involves outlier-corrupted process and measurement noise. The obtained results of all algorithms are illustrated and discussed.

5.3.1 Problem Description

We consider the problem of tracking a maneuvering target in a two-dimensional plane, for instance an aircraft that travels at a constant altitude. The state vector x_k consists of the Cartesian position and velocity. The vehicle position is available as noise-corrupted measurement. This is obviously a simplified real-life target tracking problem, but suffices to compare different filtering algorithms.

The target states are generated according to a constant velocity model, as given in Bar-Shalom et al. (2001). The target dynamics are linear with

$$x_{k+1} = Fx_k + v_k = \begin{bmatrix} I_2 & TI_2 \\ 0 & I_2 \end{bmatrix} x_k + v_k, \tag{5.24a}$$

where $T = 3$ is the sampling time in seconds, and I_2 is a two-dimensional identity matrix. The state transition matrix F is time-invariant.

The measurements are generated according to

$$y_k = Hx_k + e_k = \begin{bmatrix} I_2 & 0 \end{bmatrix} x_k + e_k. \tag{5.24b}$$

Each y_k is hence a noise-corrupted snapshot of the true target position. Also, the measurement matrix H is time-invariant.

The process and measurement noise, v_k and e_k, are subject to outliers and generated according to Gaussian mixtures,

$$v_k \sim \begin{cases} \mathcal{N}(0, Q), & \text{with probability } 0.95, \\ \mathcal{N}(0, 1000Q), & \text{with probability } 0.05, \end{cases} \tag{5.25a}$$

$$e_k \sim \begin{cases} \mathcal{N}(0, R), & \text{with probability } 0.9, \\ \mathcal{N}(0, 100R), & \text{with probability } 0.1. \end{cases} \tag{5.25b}$$

The *nominal* noise parameters are given by

$$Q = \begin{bmatrix} \frac{T^3}{3} I_2 & \frac{T^2}{2} I_2 \\ \frac{T^2}{2} I_2 & T I_2 \end{bmatrix}, \quad R = 100 I_2. \tag{5.25c}$$

Put into words, v_k and e_k are most of the times generated from nominal Gaussian distributions with nominal covariances Q and R, respectively. About five percent of the process and ten percent of the measurement noise realizations are drawn from Gaussian distributions with severely increased covariance matrices. In the tracking context, the term *clutter* is sometimes used to describe measurements that are practically plain noise.

Each experiment involves simulating (5.24) with the noise of (5.25) and a given initial state $x_0 = [100, 100, 20, 10]^{\mathrm{T}}$ for $k = 1, \ldots, 500$. In order to later run an (unfair) optimal filter on the data, we memorize at which times the outliers in v_k and e_k occur. The position trajectories of ten experiments are given in Figure 5.2. Some of the trajectories exhibit abrupt turns. Figure 5.3 illustrates the corresponding speed signals over time. These are computed from the velocity components of the state vector, and contain larger steps that can be interpreted as target maneuvers.

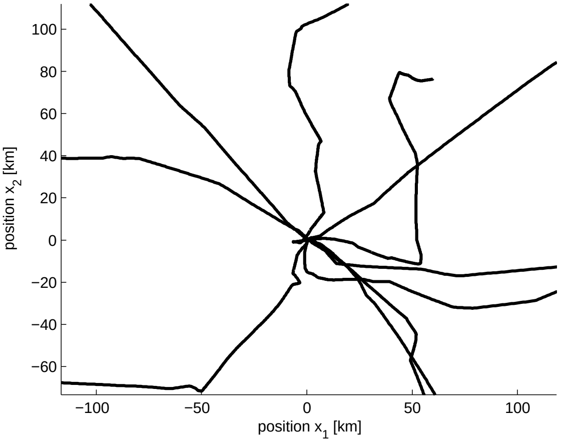


Figure 5.2: *Position trajectories for 10 different experiment realizations.*

With the noise distributions as Gaussian mixtures, the true filtering posterior $p(x_k|y_{1:k})$ is actually a Gaussian mixture density with an exponentially increasing number of components. An exact filtering algorithm based on the direct application of the Bayesian filtering recursions of Section 2.3 is hence not tractable. Furthermore, there is a mismatch between the noise in (5.25) and the densities (5.2) that we assumed in the derivation of the t filter. We specifically choose this configuration to show how the filter performs under modeling uncertainty, with merely the nominal Q and R at hand.

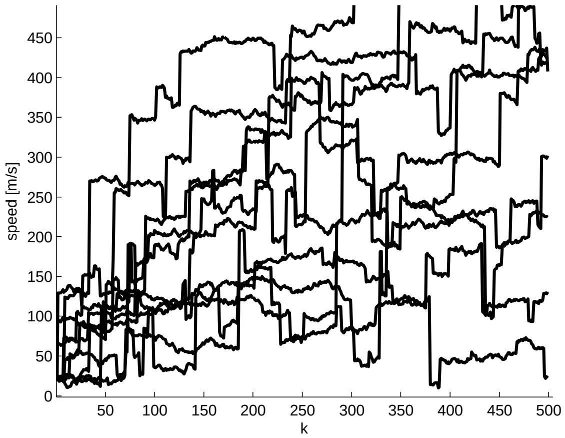


Figure 5.3: *Speed as function of time for 10 different experiment realizations.*

5.3.2 Applied Filtering Algorithms

We first note that the above tracking problem is not trivially solved by tuning of noise parameters in a Kalman filter, because both noise sources v_k and e_k contain outliers.

Six different filters are utilized to recover the target velocity and position. Some of them are given more knowledge about the true underlying model than others. The algorithms include three different Kalman filters, a particle filter, the variational Bayes filter of Agamennoni et al. (2012), and the filter that we developed in this chapter.

All of the filters initialized under the assumption that $p(x_0) = \mathcal{N}(\hat{x}_0, P_0)$ with

$$\hat{x}_0 = \begin{bmatrix} y_0 \\ 0 \end{bmatrix}, \quad P_0 = \begin{bmatrix} R & 0 \\ 0 & 50I_2 \end{bmatrix}. \tag{5.26}$$

The algorithms are detailed below:

- KF1 — a Kalman filter that is ignorant of any prospective outliers and assumes the nominal noise parameters Q and R for all k. Because the state space model (5.24) is time-invariant, the resulting Kalman filter converges to a time-invariant filter after a number of time steps.
- KF2 — a Kalman filter that is aware of the true noise covariance matrices of the mixture distributions in (5.25). With the true noise statistics known, this Kalman filter is an optimal, in the minimum variance sense, linear filter. The reader is referred to Section 3.1 for details and further references.
- KF3 — a Kalman filter that knows the instances at which outliers occur and can adjust its process and measurement noise covariance accordingly, for example by using $Q_k = 1000Q$ if an outlier occurred in the process noise

and $Q_k = Q$ otherwise. This filter solves the Bayesian filtering recursions and is the best available algorithm, but obviously under unfair premises. KF3 functions as benchmark for the given problem.

- PF — a particle filter that uses 1000 particles. Resampling is carried out when the effective number of particles falls below 600. Samples are generated by predicting particles according to (5.24) with noise realizations from the true distributions (5.25). Algorithmic details are given in Section 3.3. Again, the filter knows more than the nominal Q and R.

- VB — the variational Bayes filter of Algorithm II in Agamennoni et al. (2012). This filter is actually designed for systems with outliers in the measurement noise only, and it is interesting to investigate what happens when this condition is violated. The algorithm has a structure similar to the Kalman filter, but involves an iterative procedure in the measurement update. The reader is referred to the original article for further details. One scalar parameter h is required in the algorithm and after some tuning it is chosen as $h = 135$. Otherwise, the nominal parameters Q and R are applied.

- T — the t filter of Section 5.2.2. The algorithm is merely aware of the nominal parameters Q and R, but assumes that process and measurement noise are t distributed with degrees of freedom 3. The filter uses a process noise matrix parameter $\widetilde{Q}$ that is obtained by minimizing the Kullback-Leibler divergence between $\mathcal{N}(0, Q)$ and $\mathrm{St}(0, \widetilde{Q}, 3)$. In the same way, the parameters in the measurement noise and the initial state distributions, $\mathrm{St}(0, \widetilde{R}, 3)$ and $\mathrm{St}(\hat{x}_0, \widetilde{P}_0, 3)$, are found. The resulting state posterior is $p(x_k|y_{1:k}) = \mathrm{St}(\hat{x}_{k|k}, P_{k|k}, 3+2)$ at each time step, according to the measurement update equations (5.18), and (5.18g) in particular. In order to obtain closed form recursions, $\mathrm{St}(\hat{x}_{k|k}, P_{k|k}, 5)$ is approximated by $\mathrm{St}(\hat{x}_{k|k}, \widetilde{P}_{k|k}, 3)$ in (5.17a) of the time update.

5.3.3 Results and Discussion

All filters manage to solve the state estimation problem, but exhibit significant differences in the achieved accuracy. Once again, the reader is reminded that some of the algorithms use more knowledge than others. Actually, only KF1, VB, and T exclusively utilize the nominal model.

What makes the problem difficult is that effects of the process noise outliers and the measurement noise outliers cannot be distinguished by any filtering algorithm. After all, each filter processes a residual in some form, and that appears similar for target maneuvers and clutter measurements: larger than usual. Therefore, all filters except KF3 are subject to inevitable spikes in the position error. Figure 5.4 contains an illustration of the position error over time for KF1, KF2, KF3, and T for an illustrative time range. KF1 exhibits the slowest error decrease, KF2 is faster, and T is the fastest to decrease the position error after a spike occurred. KF2, however, is aware of the true noise covariances whereas KF1 and T merely use the nominal parameters. The quick decrease in the position error of

T can be attributed to the observation dependent update of the matrix parameter in (5.18f). Being the optimal Bayes filter, and knowing the instances at which outliers occur, KF3 performs well. VB and PF are omitted in Figure 5.4, but their performance is worse than KF1, as quantified in Table 5.1.

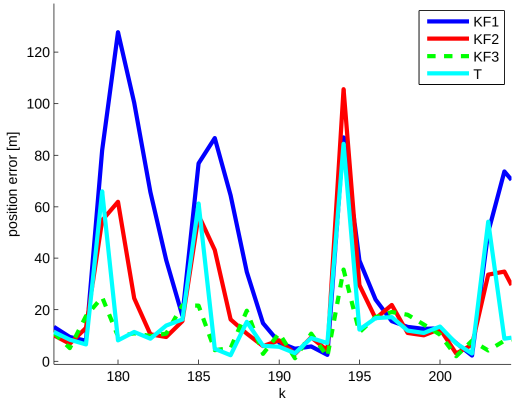


Figure 5.4: *Position error for KF1, KF2, KF3, and T, as function of time. Shown is a range of time steps of one experiment realization. All filters are subject to spikes in the position error. T is the fastest to decrease the error again. PF and VB are omitted in this illustration.*

The provided speed estimates, computed from the respective two-dimensional velocity estimates, are given in Figure 5.5 for all filters. The differences are less pronounced than in the position estimates. It appears that T and PF are a bit more agile. When no outliers act on the system, all of the algorithms appear to estimate the true speed accurately.

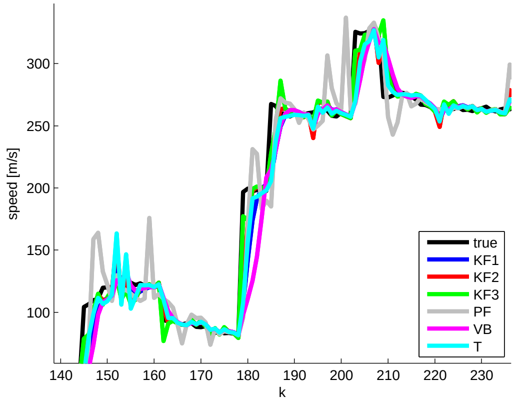


Figure 5.5: *Estimated speed by all filters. Shown is a range of time steps of one experiment realization.*

Figure 5.6 illustrates the position of all filters for four consecutive time instances. Each filter is represented by a 90 percent probability ellipse. Also illustrated are the true states, the measurements, and all particles of PF. It can be seen that the ellipses of KF1 and KF2 do not change size, because the time-invariant models that they assume makes them converge to time-invariant filters. The ellipses by KF1 and VB do not cover the true state at all. Their small ellipses indicate that both are overly confident. The particles of PF are widely spread out, and the related ellipses cover larger areas. KF2, KF3, PF, and T manage to contain the true state in their ellipses for all four time steps. KF3, being the optimal filter, is tightly fit around the true state. At $k = 23$, the ellipse of T is inflated. This can be explained by the measurement dependent matrix update (5.18f), and the fact that the measurement is on the border of the probability ellipse for $k = 22$. The decreased ellipse at $k = 24$ shows how agile the filter reacts.

So far, we have only provided qualitative insights about the algorithms. We next present results for one experiment realization which has been randomly chosen from 100 experiments. The first row of Table 5.1 contains the averaged position error, where the average is taken over all 500 time instances of the single experiment. Obviously, the optimal KF3 outperforms all its competitors. Next best is T, with mere knowledge of the nominal parameters. PF and VB are on the other end of the performance scale. Especially VB is affected by outliers in the process noise and the resulting target maneuvers. As for the maximum position error, KF1 outperforms all others. This might be explained by the fact that KF1 is the slowest to react among all filters. Rows three and four of Table 5.1 contain similar results for the speed error. Again, KF3 is followed by T. Interestingly, the difference between KF1 and KF2 is hardly visible.

Table 5.1: *Quantitative filter performance for one random experiment.*

	KF1	KF2	KF3	PF	VB	T
average position error [m]	31.9	26.1	16.8	47.7	45.2	22.2
maximum position error [m]	206.3	247.4	258.9	305.5	501.4	281.0
average speed error [m/s]	10.6	10.4	8.8	15.9	12.1	10.0
maximum speed error [m/s]	101.1	93.2	91.5	116.1	102.0	91.3

Table 5.2 also contains averaged position errors for position and speed. These are obtained by first averaging the errors over all 500 time steps, and then averaging over 100 independent experiments. The qualitative relations of Table 5.1 are preserved.

Table 5.2: *Averaged quantitative filter performance for 100 experiments.*

	KF1	KF2	KF3	PF	VB	T
average position error [m]	30.3	24.9	16.6	44.8	80.9	20.7
average speed error [m/s]	9.7	9.7	8.0	14.6	13.5	9.0

We conclude the discussion of this example by summarizing comments for each

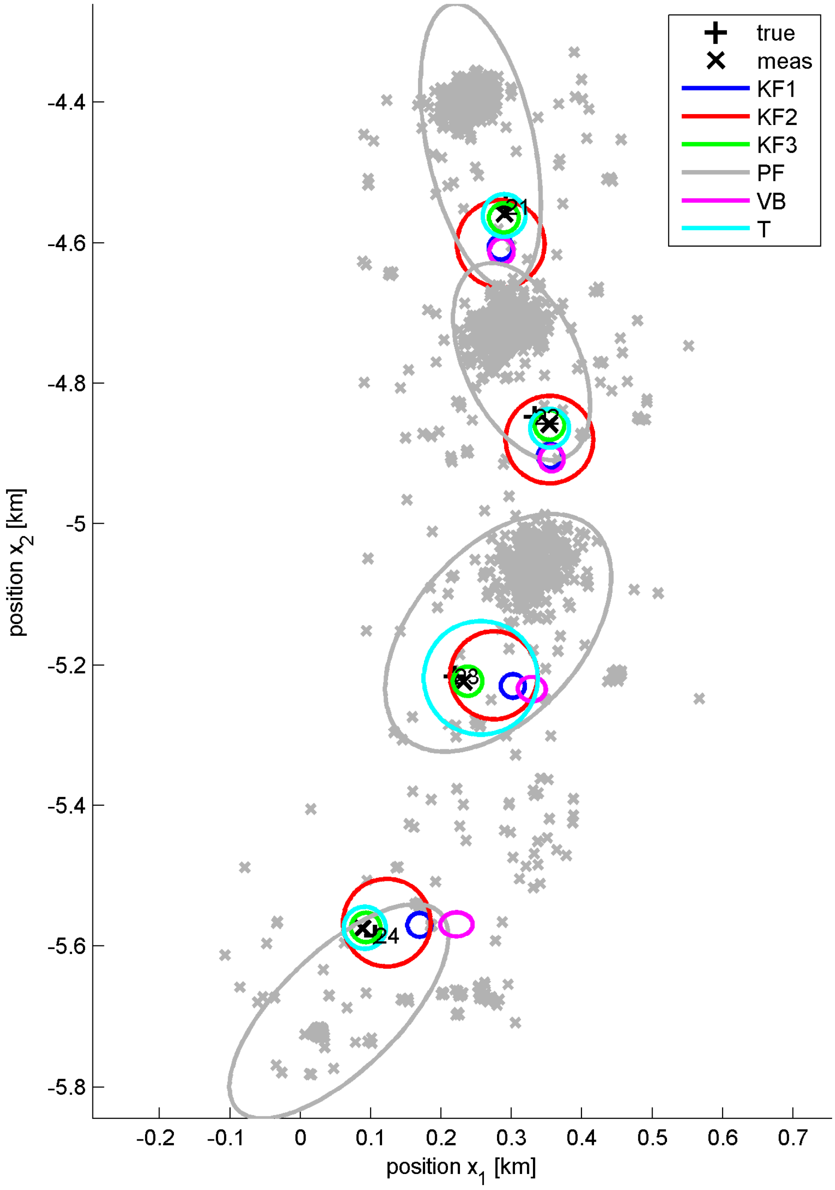


Figure 5.6: *Position estimates represented via 90 percent probability ellipses for all filters. The true state, the measurement, and all PF particles are also shown. Displayed are four consecutive time instances* $k = 21, \ldots, 24$.

filter individually:

- KF1 and KF2 — Both converge to time-invariant filters. For periods without outliers, both converge to the true state. KF2 is quicker in decreasing the estimation error.
- KF3 — The filter is optimal for the considered problem. However, it uses more information than any of the other algorithms, and knows which measurements can be trusted and which are bad. The knowledge about process and measurement noise is complementary and yields highly confident estimation results.
- PF — We restricted the number of particles to 1000. This is too low for the presented application. The performance can be improved significantly by using 10000 particles instead, of course at the price of an increased computational demand. Also, an alternative proposal distribution may improve its performance. Even with 1000 particles, all other algorithms take only a fraction of the time that PF requires.
- VB — The algorithm appears to be very sensitive to outliers in the process noise. Of course, it was derived under the premises that the process noise is well behaved, and indeed it works very well if outliers only appear in the measurement noise. The accuracy can be significantly improved by increasing the utilized process noise covariance.
- T — Given that only the nominal parameters are used, the algorithm seems robust to outliers in the process and measurement noise. It outperforms the competitors even though the underlying noise does not meet the noise assumptions that were used in the filter derivation. The chosen degrees of freedom parameter 3 makes the filter assume heavy-tailed distributions, yet with finite covariance, for all signals.

5.4 Conclusions

Based on the t distribution of Appendix A.3, we derived a filter that is a nonlinear generalization of the Kalman filter, with matrix updates that depend on the measurements. The algorithm is as simple to implement as the Kalman filter and comparable in terms of computational load. The potential application lies in problems with heavy-tailed process and measurement noise. Conceivable examples include tracking of maneuvering targets in clutter as well as problems that rely on oversimplified models of real world processes. We showed that the suggested filter outperforms state-of-the-art algorithms on a challenging example.

6 Concluding Remarks

We would like to use this final chapter to highlight some important aspects that we encountered throughout the thesis. Furthermore, some ideas for future research topics are listed.

Lessons learned

With the Bayesian filtering problem in linear Gaussian state space models solved by the Kalman filter, we addressed topics that are yet to be fully understood.

The application of Kalman filters in nonlinear models essentially requires computing mean values and covariance matrices, which are usually not available as closed form expressions. Motivated by this, we defined the *moment computation problem*, which amounts to finding the mean value and covariance matrix of a nonlinearly transformed random variable, and described a range of methods that attempt to solve it in Chapter 4. One main lesson is that none of these methods should be applied without further considerations. Monte Carlo methods, for instance, are asymptotically accurate but might not be applicable due to limited computational resources. The computationally cheap deterministic sampling schemes, in turn, have limited accuracy and are difficult to assess for arbitrary nonlinearities. A thorough analysis of the involved functions, for instance by comparison with Taylor polynomials of a certain degree, is advised.

We approached the filtering problem in linear state space models with heavy-tailed noise from an unconventional angle. The developed filter in Chapter 5 is entirely based on Student's t distribution, but requires some approximations to maintain simple recursive updates throughout time. The measurement update exhibits an interesting nonlinear structure that goes beyond that of the Kalman filter, without increasing the computational demands. Still, parts of the suggested filter clearly resemble the Kalman filter. The presented tracking example shows that the approach can yield improved results.

Future Work

A conceivable research topic that relates Section 3.2, Chapter 4, and Chapter 5 is filtering in nonlinear state space models with heavy-tailed noise. With the filter in Chapter 5 designed for linear systems, an obvious idea is linearization of the involved nonlinearities. But also the other methods in Chapter 4 might be useful in this context. Such ideas are in the spirit of the Kalman filters for nonlinear systems of Section 3.2.

In many applications, measurements are available at a high frequency. For a state estimation problem, this means that several measurements can be taken into account before an estimate needs to be computed. This corresponds to a fixed-lag smoothing problem, rather than a filtering problem. It is therefore interesting to investigate how fixed-lag variants of the algorithms in Section 3.2 can be solved with the insights that we gained in Chapter 4. Also, a smoothing extension to the algorithm in Chapter 5 is conceivable.

A

Probability Theory and Selected Distributions

A.1 Probability Densities, Expected Values, and All That

This section summarizes some useful results from probability theory. Readers that are new to the topic are referred to any introductory text on probability theory for a more extensive treatment. Good starting points are the monograph Gut (2009) or the relevant introductory chapters of Bishop (2006).

A.1.1 Remarks on Notation

We do not make a distinction between a random variable and its realization. What is meant should be obvious from context.

If x is, for example, a Gaussian random variable with parameters $\hat{x}$ and P we sometimes write $x \sim \mathcal{N}(\hat{x}, P)$. The related probability density function $p(x)$ is denoted with an extra argument as $p(x) = \mathcal{N}(x; \hat{x}, P)$. At times we use the compact notation $x \sim p(x)$ to denote that x has a distribution with density $p(x)$.

Densities carry an extra subscript where helpful: $p_x(x)$.

A.1.2 Probability Densities

The main focus of this thesis is the estimation of probability density functions, or just densities. We here list a number of basic results that are utilized throughout the entire text.

Definition of a probability density

A function $p(x)$ must satisfy $p(x) \geq 0$ everywhere on the set D and

$$\int_D p(x)\,\mathrm{d}x = 1, \tag{A.1}$$

to be a valid density. Here, D is the support of the random variable x. We shall omit D in most integral expressions for notational convenience. In many cases x is n-dimensional and real, and D is simply $\mathbb{R}^n$.

Joint densities and marginalization

A partitioned random vector $[x^\mathrm{T}, y^\mathrm{T}]^\mathrm{T}$ can be split into its components, the vectors x and y. The *marginal* density of x can be obtained by integration of the *joint* density

$$p(x) = \int p(x,y)\,\mathrm{d}y. \tag{A.2}$$

This process is known as *marginalization*.

The chain rule

The joint density $p(x,y)$ can be expressed as product of a marginal and a *conditional* density in two different ways:

$$p(x,y) = p(y|x)p(x) = p(x|y)p(y). \tag{A.3}$$

The above equation is sometimes referred to as *chain rule* of probability theory.

Bayes' theorem

By rearranging (A.3), a formula for the conditional density $p(x|y)$ can be obtained:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}. \tag{A.4}$$

Equation (A.4) is known as *Bayes' theorem* and provides the basis for the central estimation scheme in this thesis: *Bayesian filtering*.

Conditioning on a third random variable z can be included without significant changes, as in

$$p(x|y,z) = \frac{p(y|x,z)p(x|z)}{p(y|z)}. \tag{A.5}$$

Independence

Returning to jointly distributed random variables, we say that x and y are *independent* if their joint density $p(x,y)$ factors as follows

$$p(x,y) = p(x)p(y). \tag{A.6}$$

A weaker property than independence is uncorrelatedness, and discussed in the next section.

Mode and median

The *mode* of a distribution is given by $\arg\max_x p(x)$, that is the value of x which maximizes the density $p(x)$. Although $p(x)$ can have several maxima in general, the focus in this thesis is on *unimodal* distributions that have one unique "peak". Such unimodal distributions include the Gaussian, among all other elliptically contoured distributions of Section 4.2.

The *median* of a univariate distribution with density $p(x)$ is given by the value $\bar{x}$ for which

$$\int_{-\infty}^{\bar{x}} p(x)\,\mathrm{d}x = \int_{\bar{x}}^{\infty} p(x)\,\mathrm{d}x = \frac{1}{2}. \tag{A.7}$$

The median thus divides the support of x into two regions with equal probability mass.

A.1.3 Expected Values

We here mainly define the notation for certain expected values which play a central role in the text. The focus is on *mean* values and *covariance* matrices, but also *higher moments* are mentioned.

Mean value

Let x be a random vector with density $p(x)$. The *mean* of x, if it exists, is defined by the integral

$$\mathrm{E}\{x\} = \int x p(x)\,\mathrm{d}x, \tag{A.8}$$

where the integration is over the support of x.

Covariance matrices and correlation

Let y be another random vector such that x and y have the joint density $p(x, y)$. Then, the *covariance matrix* of x and y is given by

$$\operatorname{cov}\{x, y\} = \int (x - \mathrm{E}\{x\})(y - \mathrm{E}\{y\})^{\mathrm{T}} p(x, y)\,\mathrm{d}x\,\mathrm{d}y. \tag{A.9}$$

The matrix $\operatorname{cov}\{x, y\}$ indicates how a variation in x influences y, and vice versa.

If just one of the arguments is present, as in $\operatorname{cov}\{x\}$, we mean $\operatorname{cov}\{x, x\}$. Such a covariance matrix is by construction symmetric and positive semi-definite. The diagonal elements of $\operatorname{cov}\{x\}$ are the variances of all components x_i, $i = 1, \ldots, n$. If $\operatorname{cov}\{x\}$ is only positive semi-definite, as opposed to positive definite, then at least one component x_i is a fully deterministic function of other components.

If $\operatorname{cov}\{x, y\} = 0$, then x and y are *uncorrelated*. In general, this is less strict than independence of x and y. For random variables x and y that are jointly Gaussian, uncorrelatedness implies independence.

Moments

The entries in the mean vector and covariance matrix of x are so called *moments*. Let x_i, x_j, x_k be scalar components of x. Then

$$\mathrm{E}\{x_i\}, \quad \mathrm{E}\{x_i x_j\}, \quad \mathrm{E}\{x_i x_j x_k\}, \quad i,j,k = 1,\dots,n, \tag{A.10a}$$

are all first, second, and third *raw moments* of x. The quantities

$$\begin{aligned}&\mathrm{E}\{(x_i - \mathrm{E}\{x_i\})(x_j - \mathrm{E}\{x_j\})\},\\ &\mathrm{E}\{(x_i - \mathrm{E}\{x_i\})(x_j - \mathrm{E}\{x_j\})(x_k - \mathrm{E}\{x_k\})\}, \quad i,j,k = 1,\dots,n,\end{aligned} \tag{A.10b}$$

are all second and third *central moments* of x. Obviously, $\mathrm{cov}\{x\}$ is composed of central second moments.

For a distribution as simple as the Gaussian, all moments of $x \sim \mathcal{N}(\hat{x}, P)$ are readily available as functions of the entries in $\hat{x}$ and P. The formulas are, for instance, given in Papoulis and Pillai (2002). Other distributions have to fulfill some conditions on their parameters for certain moments to exist. Examples include Student's t or the F distribution that are covered in Appendix A.3 and Appendix A.2.2, respectively.

The number of moments increases tremendously with the dimension of x. For an n-vector x there are n first, $n(n+1)/2$ distinct second, and $n(n+1)(n+2)/6$ distinct third moments to compute.

Expected values of transformed random variables

A central element in this thesis are expected values of $y = f(x)$, some function of the random vector x. For some cases the density of y is known. The task of computing $\mathrm{E}\{f(x)\}$ can then be reduced to an integral of the form (A.8). Alternatively, we need to integrate with respect to the density of x as in

$$\mathrm{E}\{f(x)\} = \int f(x) p(x)\,\mathrm{d}x. \tag{A.11}$$

In a similar way, we can compute the covariance matrix of two functions $f_1(x)$ and $f_2(x)$ as follows

$$\mathrm{cov}\{f_1(x), f_2(x)\} = \int (f_1(x) - \mathrm{E}\{f_1(x)\})(f_2(x) - \mathrm{E}\{f_2(x)\})^{\mathrm{T}} p(x)\,\mathrm{d}x. \tag{A.12}$$

Approximate computation of the above two moment integrals is the core task in Chapter 4.

Conditional expected values

Extending the idea of conditional densities, we also define conditional expected values. Here, $\mathrm{E}\{x|y\}$ is to be understood as

$$\mathrm{E}\{x|y\} = \int x p(x|y)\,\mathrm{d}x. \tag{A.13}$$

Similarly, $\mathrm{cov}\{x|y\}$ denotes a covariance matrix where the integration involves the conditional density.

A.1.4 The Transformation Theorem

We here introduce a powerful tool which we refer to as the *transformation theorem*. It can be used to find the density of a random variable that is transformed by a function. We only present the theorem in its most basic form, but refer the reader to Rao (2001) or Gut (2009) for further details. For the sake of clarity, all probability density functions carry an extra index.

Transformation theorem: Let x be a scalar random variable with density $p_x(x)$, where x admits values from its support D_x. Let f in $y = f(x)$ be a one-to-one mapping for which an inverse g with $x = g(y)$ exists, with an existing derivative $g'(y)$ that is continuous. Then, the random variable $y = f(x)$ has the density

$$p_y(y) = \begin{cases} p_x(g(y))|g'(y)|, & y \in D_y, \\ 0, & \text{otherwise}, \end{cases} \tag{A.14}$$

with $D_y = \{f(x) : x \in D_x\}$.

The theorem is presented in Chapter 3 of Rao (2001) and Chapter 1 of Gut (2009), including proofs and extensions to cases when $f(x)$ is not a one-to-one mapping (bijection). The occurrence of $|g'(y)|$ is the result of a change of integration variables, and in the multivariate case $g'(y)$ is merely replaced by the Jacobian matrix of $g(y)$. In case when y is of lower dimension than x, auxiliary variables need to be introduced and subsequently marginalized. An example of such a procedure is given for the derivation of the multivariate t density in Appendix A.3.6.

A.2 Selected Probability Distributions

We here present some well known probability distributions that are used in the thesis. All of these are covered in a plethora of text books. We adopt the presentation of Bishop (2006) and DeGroot (2004).

A.2.1 The Gaussian Distribution

An n-dimensional random variable x with Gaussian distribution $\mathcal{N}(\hat{x}, P)$ is fully characterized by its mean $\hat{x}$ and covariance matrix P. For a positive definite matrix P, the density is given by

$$\mathcal{N}(x; \hat{x}, P) = \frac{1}{(2\pi)^{\frac{n}{2}}} \frac{1}{\sqrt{\det(P)}} \exp\left(-\frac{1}{2}(x-\hat{x})^{\mathrm{T}} P^{-1} (x-\hat{x})\right). \tag{A.15}$$

Information about the many convenient properties of the Gaussian distribution can be found in many references. A particularly extensive account is given in Bishop (2006). We shall therefore only briefly mention some aspects. Gaussian random variables that are uncorrelated are also independent. The marginal and conditional distribution of a partitioned Gaussian random vector are again Gaussian, which relates to the Gaussian distribution being its own conjugate prior for the mean in Bayesian statistics. The formulas for marginal and conditional density can be used to derive the Kalman filter from a Bayesian perspective, as seen in Anderson and Moore (1979).

In the following paragraphs, we briefly list some of the most important results without derivation.

If $x \sim \mathcal{N}(\hat{x}, P)$, then $Ax + b \sim \mathcal{N}(A\hat{x} + b, APA^{\mathrm{T}})$.

From the joint density of two Gaussian random vectors x_1 and x_2,

$$p(x_1, x_2) = \mathcal{N}(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}; \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}, \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^{\mathrm{T}} & P_{22} \end{bmatrix}), \tag{A.16}$$

follows that the marginal density of x_1 is given by

$$p(x_1) = \mathcal{N}(x_1; \hat{x}_1, P_{11}). \tag{A.17}$$

The conditional density of $x_1|x_2$ is given by

$$p(x_1|x_2) = \mathcal{N}(x_1; \hat{x}_1 + P_{12}P_{22}^{-1}(x_2 - \hat{x}_2), P_{11} - P_{12}P_{22}^{-1}P_{12}^{\mathrm{T}}). \tag{A.18}$$

A.2.2 Useful Univariate Distributions

The Gamma distribution

A scalar random variable x admits the Gamma distribution $\mathrm{Gam}(\alpha, \beta)$ with shape parameter $\alpha > 0$ and rate (or inverse scale) parameter $\beta > 0$ if it has the density

$$\mathrm{Gam}(x; \alpha, \beta) = \begin{cases} \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x), & x > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{A.19}$$

Here, $\Gamma(z) = \int_0^{\infty} e^{-t} t^{z-1} \, \mathrm{d}t$ is the Gamma function. Two parametrizations of the Gamma distribution exist in literature: the rate version that we adopt, and a version that uses the inverse rate (called scale parameter) instead.

Using the transformation theorem of Appendix A.1.4 with $x \sim \mathrm{Gam}(\alpha, \beta)$, we can derive that $cx \sim \mathrm{Gam}(\alpha, \beta/c)$ for some positive scalar c. Mean and variance of $x \sim \mathrm{Gam}(\alpha, \beta)$ are $\mathrm{E}\{x\} = \alpha/\beta$ and $\mathrm{var}\{x\} = \alpha/\beta^2$, respectively. For $\alpha = \beta \to \infty$ the support of the density collapses and $\mathrm{Gam}(x; \alpha, \beta)$ approaches a Dirac impulse centered at one.

The chi-squared distribution

A special case of the Gamma distribution for $\alpha = \nu/2$, $\beta = 1/2$ is the chi-squared distribution $\chi^2(\nu)$ with degrees of freedom $\nu > 0$ and density

$$\chi^2(x; \nu) = \mathrm{Gam}(x; \frac{\nu}{2}, \frac{1}{2}) = \begin{cases} \frac{2^{-\frac{\nu}{2}}}{\Gamma(\frac{\nu}{2})} x^{\frac{\nu}{2}-1} \exp(-\frac{1}{2}x), & x > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{A.20}$$

From the scaling property of the Gamma distribution, we have that from $x \sim \chi^2(\nu)$ follows $x/\nu \sim \mathrm{Gam}(\frac{\nu}{2}, \frac{\nu}{2})$. This relates to a common generation rule for t random variables, which are treated in Appendix A.3.

The inverse Gamma distribution

In order to compute the moments of t random vectors, it is required to work with the inverse of a Gamma random variable. We shall here present the related density, and even sketch the derivation as it is not contained in Bishop (2006) or DeGroot (2004).

Let $y \sim \text{Gam}(\alpha, \beta)$. Then its inverse $x = 1/y$ has an inverse Gamma distribution with parameters $\alpha > 0$ and $\beta > 0$, and density

$$\text{IGam}(x; \alpha, \beta) = \begin{cases} \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp(-\beta/x), & x > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{A.21}$$

The result follows from the transformation theorem of Appendix A.1.4. Using $g(x) = 1/x$, we obtain

$$\begin{aligned} p_X(x) &= p_Y(g(x))|g'(x)| \\ &= \text{Gam}(x^{-1}; \alpha, \beta)x^{-2}, \end{aligned}$$

which gives (A.21). The density is for one set of parameters illustrated in Figure A.1.

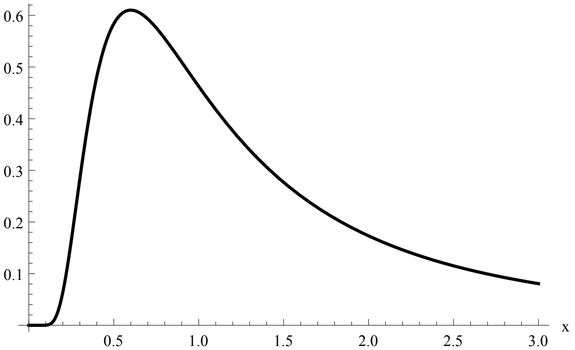


Figure A.1: *The density of an inverted Gamma random variable for one set of parameters,* $\text{IGam}(x; 1.5, 1.5)$.

The mean of $x \sim \text{IGam}(\alpha, \beta)$ is $\text{E}\{x\} = \beta/(\alpha - 1)$ and is finite for $\alpha > 1$. Similar conditions are required for higher moments to exist.

The F distribution

We conclude this presentation of univariate densities with the F distribution. Consider a scaled ratio of two chi-squared random variables $x \sim \chi^2(a)$ and $y \sim \chi^2(b)$,

$$z = \frac{x/a}{y/b}.$$

We recall that this is also a ratio of two Gamma random variables. The random variable z then admits an F distribution with $a > 0$ and $b > 0$ degrees of freedom. Its density is given by

$$\mathcal{F}(z;a,b) = \begin{cases} \frac{\Gamma(\frac{a+b}{2})}{\Gamma(\frac{a}{2})\Gamma(\frac{b}{2})} a^{\frac{a}{2}} b^{\frac{b}{2}} \frac{z^{\frac{a}{2}-1}}{(b+az)^{\frac{a+b}{2}}}, & z > 0, \\ 0, & \text{otherwise}. \end{cases} \tag{A.22}$$

An illustration of the density for one set of parameters is given in Figure A.2.

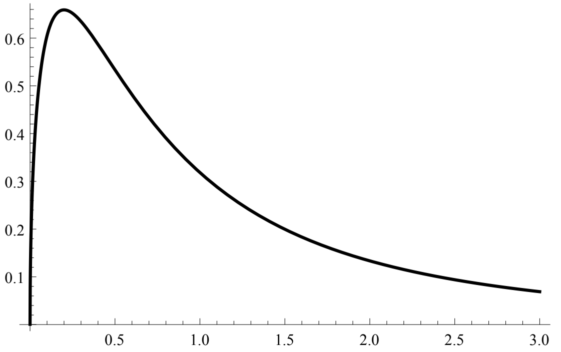


Figure A.2: *The density of a scaled ratio of chi-squared random variables for one set of parameters, $\mathcal{F}(x;3,3)$.*

The mean value of $z \sim \mathcal{F}(a,b)$ is $\mathrm{E}\{z\} = b/(b-2)$, and exists for $b > 2$. The mode, that is the maximum of $\mathcal{F}(z;a,b)$, is located at $\frac{(a-2)b}{a(b+2)}$, for $a > 2$ and $b > 2$.

A.3 Student's *t* distribution

This section summarizes a number of results for the multivariate t distribution which can exhibit heavier tails than the Gaussian distribution. It plays the central role in the filter development of Chapter 5, where it is used to model process and measurement noise outliers. For those reader that are exclusively interested in the t distribution, a technical report has been written up and published as Roth (2013).

In the following sections, it is shown how t random variables can be generated, the probability density function is derived, and marginal and conditional densities of partitioned t random vectors are presented. Moreover, a brief comparison with the multivariate Gaussian distribution is provided. The derivations of several results are given in dedicated sections.

The main references include DeGroot (2004) and Bishop (2006), which contain dedicated sections, as well as the monograph Kotz and Nadarajah (2004).

A t random variable x is fully characterized by its degrees of freedom parameter

ν (scalar and positive), a mean $\hat{x}$ (an n-vector), and a symmetric matrix parameter P (an $n \times n$-matrix). In general, P is not the covariance of x. In case P is positive definite, we can write the t density as

$$\mathrm{St}(x; \hat{x}, P, \nu) = \frac{\Gamma(\frac{\nu+n}{2})}{\Gamma(\frac{\nu}{2})} \frac{1}{(\nu\pi)^{\frac{n}{2}}} \frac{1}{\sqrt{\det(P)}} \left(1 + \frac{1}{\nu}(x - \hat{x})^{\mathrm{T}} P^{-1} (x - \hat{x})\right)^{-\frac{n+\nu}{2}}. \tag{A.23}$$

A.3.1 Representation and Moments

t random vectors can be generated in a number of ways, as summarized in Kotz and Nadarajah (2004). We here combine Gamma and Gaussian random variables and show some properties of the resulting quantity without specifying its distribution. The fact that the derived random variable admits indeed a t distribution with density (A.23) is postponed to Appendix A.3.2. The use of the more general Gamma distribution instead of a chi-squared distribution, as in DeGroot (2004), turns out to be beneficial in deriving more involved results in Appendix A.3.8.

Let $v > 0$ be a random variable that admits the Gamma distribution $\mathrm{Gam}(\alpha, \beta)$ with shape parameter $\alpha > 0$, and rate (or inverse scale) parameter $\beta > 0$. The related density is given in (A.19). Furthermore, consider the n-dimensional, zero mean, random variable y with Gaussian distribution $\mathcal{N}(0, P)$. P is the positive definite covariance matrix of y. The related density is shown in (A.15). Given that v and y are independent of one another, the following random variable is constructed

$$x = \hat{x} + \frac{1}{\sqrt{v}} y, \tag{A.24}$$

with an n-vector $\hat{x}$. Realizations of x are scattered around $\hat{x}$. The spread is intuitively influenced by the density of v, and it can be easily seen that $x|v \sim \mathcal{N}(\hat{x}, \frac{1}{v}P)$, a Gaussian with scaled covariance matrix.

In the limit $\alpha = \beta \to \infty$, the Gamma density (A.19) approaches a Dirac impulse centered at one, which results in $v = 1$ with probability 1. It follows (without further specification of the distribution) that x in (A.24) then reduces to a Gaussian random variable.

With mere knowledge of (A.24) we can derive moments of x. Trivially taking the expectation reveals $\mathrm{E}\{x\} = \hat{x}$. For higher moments, it is convenient to work with $w = \frac{1}{v}$, which admits an inverse Gamma distribution $\mathrm{IGam}(\alpha, \beta)$. The random variable w has the density (A.21) and the expected value $\mathrm{E}\{w\} = \beta/(\alpha - 1)$ for $\alpha > 1$. The covariance computations

$$\begin{aligned} \mathrm{cov}\{x\} &= \mathrm{E}(x - \hat{x})(x - \hat{x})^{\mathrm{T}} \\ &= \mathrm{E}(wyy^{\mathrm{T}}) \\ &= \frac{\beta}{\alpha - 1} P, \quad \alpha > 1, \end{aligned} \tag{A.25}$$

reveal that P is in general not a covariance matrix, and finite only if $\alpha > 1$. Higher moments can in principle be computed in a similar manner, again with conditions for existence.

A.3.2 The *t* Probability Density Function

We immediately present the main result but show detailed calculations in Appendix A.3.6.

Let $v \sim \mathrm{Gam}(\alpha, \beta)$ and $y \sim \mathcal{N}(0, P)$. Then $x = \hat{x} + v^{-\frac{1}{2}} y$ admits a t distribution with density

$$p(x) = \frac{\Gamma(\alpha + \frac{n}{2})}{\Gamma(\alpha)} \frac{1}{(2\beta\pi)^{\frac{n}{2}}} \frac{1}{\sqrt{\det(P)}} \left(1 + \frac{\Delta^2}{2\beta}\right)^{-\frac{n}{2}-\alpha} \tag{A.26a}$$

$$= \mathrm{St}(x; \hat{x}, \frac{\beta}{\alpha} P, 2\alpha), \tag{A.26b}$$

where $\Delta^2 = (x - \hat{x})^{\mathrm{T}} P^{-1} (x - \hat{x})$.

Obviously, (A.26a) is a t density in disguise. For $\alpha = \beta = \frac{\nu}{2}$, (A.26b) turns into the familiar (A.23). Intermediate steps to obtain (A.26a) and (A.26b) include introduction of an auxiliary variable and subsequent marginalization. Intermediate results reveal that β and P only appear as product βP. Consequently, β can be set to any positive number without altering (A.26b), if the change is compensated for by altering P. This relates to the fact that t random variables can be constructed using a wide range of rules. The parametrization using α and β is used to derive the conditional density of a partitioned t random variable in Appendix A.3.8.

A.3.3 Affine Transformations and Marginal Densities

The following property is common to both t and Gaussian distribution. Consider a $n_z \times n_x$-matrix A and an n_z-vector b. Let x be a t random variable with density (A.23). Then, the random variable $z = Ax + b$ has the density

$$p(z) = \mathrm{St}(z; A\hat{x} + b, APA^{\mathrm{T}}, \nu). \tag{A.27}$$

The result follows from (A.24) and the formulas for affine transformations of Gaussian random variables, that are summarized in Bishop (2006) for example. Here, A must be such that APA^{T} is invertible for the density to exist. The degrees of freedom parameter ν is not altered by affine transformations.

There is an immediate consequence for partitioned t vectors $x^{\mathrm{T}} = [x_1^{\mathrm{T}}, x_2^{\mathrm{T}}]$ that are composed of x_1 and x_2, with respective dimensions n_1 and n_2. From the joint density

$$p(x) = p(x_1, x_2) = \mathrm{St}(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}; \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}, \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^{\mathrm{T}} & P_{22} \end{bmatrix}, \nu), \tag{A.28}$$

the marginal density of either x_1 or x_2 is easily obtained by applying a linear transformation. Choosing $A = \begin{bmatrix} 0 & I \end{bmatrix}$ yields the marginal density of x_2:

$$p(x_2) = \mathrm{St}\left(x_2; \hat{x}_2, P_{22}, \nu\right). \tag{A.29}$$

A.3.4 The Conditional Density

The main result of this section is that $x_1|x_2$ admits another t distribution if x_1 and x_2 are jointly t distributed with density (A.28). The derivations are straightforward but somewhat involved, and therefore provided in Appendix A.3.8. The conditional density is given by

$$p(x_1|x_2) = \frac{p(x_1, x_2)}{p(x_2)} = \mathrm{St}(x_1; \hat{x}_{1|2}, P_{1|2}, \nu_{1|2}), \tag{A.30a}$$

where we have introduced

$$\nu_{1|2} = \nu + n_2, \tag{A.30b}$$

$$\hat{x}_{1|2} = \hat{x}_1 + P_{12}P_{22}^{-1}(x_2 - \hat{x}_2), \tag{A.30c}$$

$$P_{1|2} = \frac{\nu + (x_2 - \hat{x}_2)^{\mathrm{T}} P_{22}^{-1}(x_2 - \hat{x}_2)}{\nu + n_2}(P_{11} - P_{12}P_{22}^{-1}P_{12}^{\mathrm{T}}). \tag{A.30d}$$

These expressions are in accordance with the formulas given in Koop (2003) (without derivation) and DeGroot (2004) (derivation is given as an exercise). The book Kotz and Nadarajah (2004) misleadingly states that the conditional distribution is in general not t. However, a "generalized" t distribution is presented in Chapter 5 of Kotz and Nadarajah (2004), which provides the property that the conditional distribution is again t. The derivations given here show that there is no need to consider a generalization of the t for such a closedness property.

Obviously, conditioning increases the degrees of freedom. The expression for $\hat{x}_{1|2}$ is the same as the conditional mean in the Gaussian case. The matrix parameter $P_{1|2}$, in contrast to the Gaussian case, is also influenced by the realization x_2. By letting ν go to infinity, we can recover the Gaussian conditional covariance

$$\lim_{\nu \to \infty} P_{1|2} = P_{11} - P_{12}P_{22}^{-1}P_{12}^{\mathrm{T}}. \tag{A.31}$$

A.3.5 A Comparison with the Gaussian Distribution

The Gaussian distribution of Section A.2.1 appears to be the most popular distribution among continuous multivariate unimodal distributions. Motivation for this is given by the many convenient properties: simple marginalization, conditioning, parameter estimation from data. We here compare the t distribution with the Gaussian to assess whether some of the known Gaussian results generalize. After all, the t distribution can be interpreted as generalization of the Gaussian (or conversely, the Gaussian as special case of the t distribution).

Gaussian distribution as limit case

It is a well known result that the t distribution converges to a Gaussian distribution as the degrees of freedom ν tend to infinity, see Bishop (2006). This fact can be established, for instance, using (A.24) and the properties of the Gamma distribution. Another encounter with this result is described in Appendix A.3.6.

Tail behavior

One reason to employ the t distribution is its ability to generate *outliers*, which facilitates more accurate modeling of many real world scenarios. In contrast to the Gaussian, the t distribution can exhibit *heavy tails*. To illustrate this, we consider the 2-dimensional example with parameters

$$\hat{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & -0.4 \\ -0.4 & 1 \end{bmatrix}, \quad \nu = 5. \tag{A.32}$$

Figure A.3 contains 25 samples each, drawn from the Gaussian $\mathcal{N}(\hat{x}, P)$ and the t distribution $\mathrm{St}(\hat{x}, P, \nu)$. Close to $\hat{x}$, there is no obvious difference between Gaussian and t samples. More t samples are scattered further from the mean, with one outlier being far off. The illustrated phenomenon can be explained by the t distribution having heavier tails: the density does not decay as quickly when moving away from the mean.

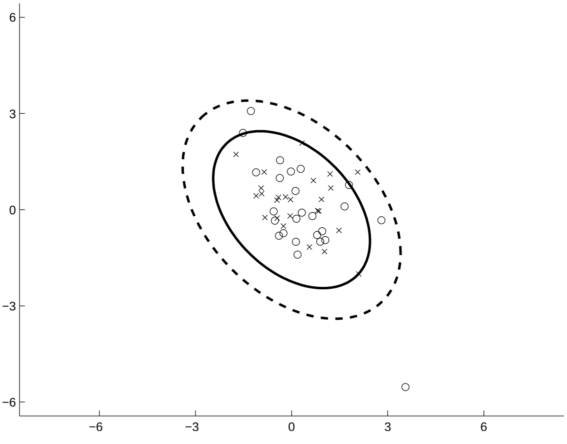
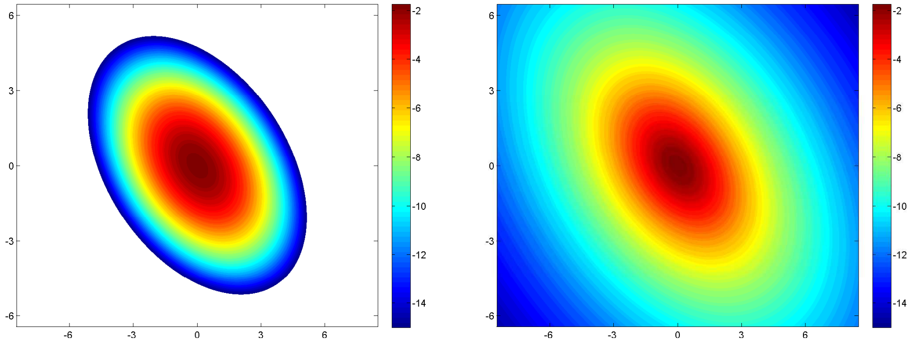


Figure A.3: 25 random samples each from the 2-dimensional distributions $\mathcal{N}(\hat{x}, P)$ (crosses) and $\mathrm{St}(\hat{x}, P, \nu)$ (circles) with the parameters given in (A.32). *The ellipses illustrate regions that contain 95 percent of the probability mass, solid for the Gaussian and dashed for the* t *distribution.*

For the presented two-dimensional case, we can compute both Gaussian and t density on a grid and visualize the results. Figure A.4 contains heat maps that illustrate the density logarithms $\ln \mathcal{N}(x; \hat{x}, P)$ and $\ln \mathrm{St}(x; \hat{x}, P, \nu)$, where the logarithmic scale helps highlight the difference in tail behavior. The coloring is the same in Figure A.4a and Figure A.4b, the former contains a large proportion of white ink which corresponds to values below -15. While the densities appear similar close to $\hat{x}$, the Gaussian drops quickly with increasing distance whereas the t density shows a less rapid decay. The connection between the occurrence of samples that lie on ellipses that are further away from $\hat{x}$ and the corresponding values of the density is apparent.



(a) $\ln \mathcal{N}(x; \hat{x}, P)$.

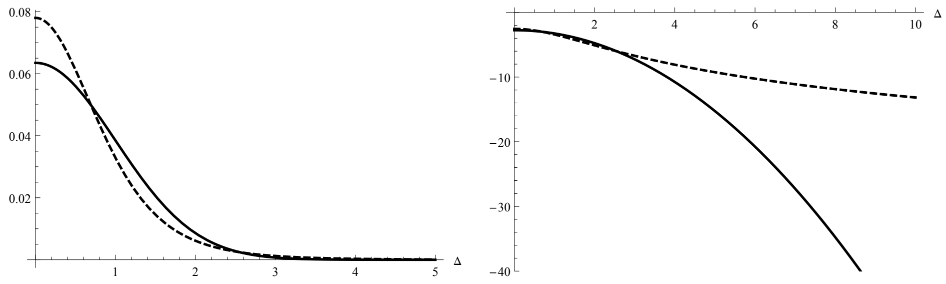
(b) $\ln \mathrm{St}(x; \hat{x}, P, \nu)$.

Figure A.4: *Heat maps of* $\ln \mathcal{N}(x; \hat{x}, P)$ *and* $\ln \mathrm{St}(x; \hat{x}, P, \nu)$ *for a two-dimensional* x*. The logarithmic presentation helps highlight the difference in tail behavior. White areas in (a) correspond to values below* -15*. The parameters are given by* (A.32).

The general n-dimensional case is a bit trickier to illustrate. An investigation of the density expression (A.23) and (A.15), however, reveals that both can be written as functions of a scalar quantity $\Delta^2 = (x - \hat{x})^{\mathrm{T}} P^{-1} (x - \hat{x})$. A constant Δ^2 corresponds to an ellipsoidal region (an ellipse for $n = 2$) with constant density, as seen in Figure A.4. Δ^2 can be interpreted as a squared weighted distance and is widely known as the *squared Mahalanobis distance*, in for instance Bishop (2006). Having understood the distance dependence, it is reasonable to plot each density as a function of Δ. Furthermore, P can be set to unity without loss of generality when comparing $\mathcal{N}(x; \hat{x}, P)$ and $\mathrm{St}(x; \hat{x}, P, \nu)$, because of the common factor $\det(P)^{-1/2}$. Figure A.5 illustrates the Gaussian and t density as functions of Δ with $P = I_n$. The shape of the curves is determined by ν and n only. For $n = 1$, such illustrations coincide with the univariate densities (over the positive ray). Here, we display the functions for $n = 3$ and $\nu = 3$. For the chosen parameters, the t density appears more peaked towards the mean (small Δ in Figure A.5a). The logarithmic display (Figure A.5b) again shows the less rapid decay of the t density with increasing Δ, commonly known as heavy tails.

Dependence and correlation

Gaussian random variables that are uncorrelated are by default independent: the joint density is the product of marginal densities. This is not the case for arbitrary other distributions, as we shall now establish for the t distribution. Consider the



***(a)** Linear scale.*

***(b)** Logarithmic scale.*

***Figure A.5:** Gaussian density (solid) and* t *density (dashed) with $n = 3$ and $\nu = 3$, as functions of the Mahalanobis distance Δ.*

a partitioned t random vector x with density (A.28). For $P_{12} = 0$ we see from the covariance result (A.25) that x_1 and x_2 are uncorrelated. The joint density

$$p(x_1, x_2) \propto \left(1 + \frac{1}{\nu}(x_1 - \hat{x}_1)^{\mathrm{T}} P_{11}^{-1}(x_1 - \hat{x}_1) + \frac{1}{\nu}(x_2 - \hat{x}_2)^{\mathrm{T}} P_{22}^{-1}(x_2 - \hat{x}_2)\right)^{-\frac{n}{2} - \frac{\nu}{2}}$$

does not factor trivially, however. Consequently, uncorrelated t random variables are in general not independent. Furthermore, the product of two t densities is in general not a t density (unless we have $p(x_1|x_2)p(x_2)$).

Probability regions and distributions of quadratic forms

The following paragraphs make extensive use of the results in Appendix A.2.

For a Gaussian random variable $x \sim \mathcal{N}(\hat{x}, P)$, it is well known that the quadratic form (or squared Mahalanobis distance) $\Delta^2 = (x - \hat{x})^{\mathrm{T}} P^{-1}(x - \hat{x})$ admits a chi-squared distribution $\chi^2(n)$. A derivation is given as example in Section 4.2. Here, we use this result to define symmetric regions around $\hat{x}$ in which a fraction, say α percent, of the probability mass is concentrated. Such an α-percent probability region contains all x for which

$$(x - \hat{x})^{\mathrm{T}} P^{-1}(x - \hat{x}) \leq \Delta_\alpha^2, \quad \text{with} \quad \int_0^{\Delta_\alpha^2} \chi^2(r; n)\, \mathrm{d}r = \alpha/100. \qquad \text{(A.33)}$$

The required Δ_α^2 can be obtained from the *inverse cumulative distribution function* of $\chi^2(n)$, which is implemented in many software packages.

For $x \sim \mathrm{St}(\hat{x}, P, \nu)$ similar statements can be made. The computations, however, are slightly more involved. Recall from (A.24) that x can be generated using a Gaussian random vector $y \sim \mathcal{N}(0, P)$ and the Gamma random variable $v \sim \mathrm{Gam}(\nu/2, \nu/2)$. Furthermore, observe that v can be written as $v = w/\nu$ with a chi-squared random variable $w \sim \chi^2(\nu)$, as explained in Appendix A.2. Using

$$x = \hat{x} + \frac{1}{\sqrt{v}} y = \hat{x} + \frac{1}{\sqrt{w/\nu}} y, \tag{A.34}$$

we can re-arrange the squared form

$$\Delta^2 = (x - \hat{x})^{\mathrm{T}} P^{-1} (x - \hat{x}) = n \frac{y^{\mathrm{T}} P^{-1} y / n}{w/\nu}, \tag{A.35}$$

and obtain a ratio of two scaled chi-squared variables, since $y^{\mathrm{T}} P^{-1} y \sim \chi^2(n)$. Knowing that such a ratio admits an F-distribution, we have

$$\frac{1}{n} (x - \hat{x})^{\mathrm{T}} P^{-1} (x - \hat{x}) \sim \mathcal{F}(n, \nu), \tag{A.36}$$

with density given by (A.22). An alternative, more compact derivation of this result is provided as example in 4.2. Following the steps that were taken for the Gaussian case above, we can derive a symmetric α-percent probability region for $x \sim \mathrm{St}(\hat{x}, P, \nu)$ as the set of all x for which

$$(x - \hat{x})^{\mathrm{T}} P^{-1} (x - \hat{x}) \leq n r_\alpha, \quad \text{with} \quad \int_0^{r_\alpha} \mathcal{F}(r; n, \nu) \, \mathrm{d}r = \alpha/100. \tag{A.37}$$

Again, the required r_α can be obtained from an inverse cumulative distribution function that is available in many software packages.

Figure A.3 contains two ellipses, a solid for the Gaussian and a dashed for the t distribution. Both illustrate regions which contain 95 percent of the probability mass, and have been obtained with the techniques explained in this section. It can be seen that the 95 percent ellipse for the t distribution stretches out further than the Gaussian ellipse, which of course relates to the heavy tails of the t distribution.

Parameter Estimation from Data

Methods for finding the parameters of a Gaussian random variables from a number of independent and identically distributed realizations are well known. A comprehensive treatment of *maximum likelihood* and *Bayesian* methods is given in Bishop (2006). The solutions are available in closed form and comparably simple. The maximum likelihood estimate for the mean of a Gaussian, for instance, coincides with the average over all realizations.

For the t distribution there are no such simple closed form solutions. However, iterative maximum likelihood schemes based on the *expectation maximization* algorithm are available. The reader is referred to McLachlan and Krishnan (2008) for further details.

A.3.6 Derivation of the *t* Density

Here, we make use of the transformation theorem of Appendix A.1.4 to derive the density of the random vector in (A.24). This technique involves the introduction of an auxiliary variable, a change of integration variables, and subsequent marginalization of the auxiliary variable. In the following calculations, we merely require (A.24) and the joint density of v and y.

As v and y are independent, their joint density is the product of individual densities

$$p_{v,y}(v,y) = p_v(v)p_y(y).$$

We now introduce the pair u and x, which are related to v and y through

$$x = \hat{x} + \frac{1}{\sqrt{v}}y; \quad u = v.$$

Here, it is x that we are actually interested in. u is an auxiliary variable that we have to introduce in order to get a one to one relation between (u,x) and (v,y):

$$y = \sqrt{u}(x - \hat{x}); \quad v = u.$$

According to the transformation theorem of Appendix A.1.4, the density of (u,x) is given by

$$p_{u,x}(u,x) = p_v(u)p_y\left(\sqrt{u}(x-\hat{x})\right)|\det(J)|, \tag{A.38}$$

with J as the Jacobian of the variable change relation

$$J = \begin{bmatrix} \partial y/\partial x^{\mathrm{T}} & \partial y/\partial u \\ \partial v/\partial x^{\mathrm{T}} & \partial v/\partial u \end{bmatrix} = \begin{bmatrix} \sqrt{u}I_n & \frac{1}{2\sqrt{u}}(x-\hat{x}) \\ 0 & 1 \end{bmatrix}.$$

With J being upper triangular, $\det(J)$ simplifies to $\det(J) = \det(\sqrt{u}I_n) = u^{\frac{n}{2}}$. We next write down each factor of the joint density (A.38)

$$\begin{aligned} p_y(\sqrt{u}(x-\hat{x})) &= \frac{1}{(2\pi)^{\frac{n}{2}}}\frac{1}{\sqrt{\det(P)}}\exp\left(-\frac{1}{2}\sqrt{u}(x-\hat{x})^{\mathrm{T}}P^{-1}\sqrt{u}(x-\hat{x})\right) \\ &= u^{-\frac{n}{2}}\mathcal{N}(x;\hat{x},P/u), \\ p_v(u) &= \mathrm{Gam}(u;\alpha,\beta), \end{aligned}$$

and combine them according to (A.38) to obtain the joint density

$$p_{u,x}(u,x) = \mathrm{Gam}(u;\alpha,\beta)\mathcal{N}(x;\hat{x},P/u).$$

From the above expression, we can now integrate out u to obtain

$$p_x(x) = \int_0^\infty \mathrm{Gam}(u;\alpha,\beta)\mathcal{N}(x;\hat{x},P/u)\,\mathrm{d}u. \tag{A.39}$$

The alert reader might at this stage recognize the integral representation of the t density, given in Bishop (2006), for the case $\alpha = \beta = \frac{\nu}{2}$. As we proceed further and compute the integral, the connections become more evident. Also, we see that if the Gamma part reduces to a Dirac impulse $\delta(u-1)$ for $\alpha = \beta \to \infty$, the result of the integration is a Gaussian density.

To simplify the upcoming calculations we introduce the squared Mahalanobis distance $\Delta^2 = (x - \hat{x})^{\mathrm{T}} P^{-1} (x - \hat{x})$ and proceed:

$$\begin{aligned}
p_X(x) &= \int_0^\infty \frac{\beta^\alpha}{\Gamma(\alpha)} u^{\alpha-1} \exp(-\beta u) \frac{1}{(2\pi)^{\frac{n}{2}}} \frac{1}{\sqrt{\det(P/u)}} \exp\left(-\frac{u}{2}\Delta^2\right) \mathrm{d}u \\
&\propto \int_0^\infty u^{\frac{n}{2}+\alpha-1} \exp\left(-u(\beta + \frac{\Delta^2}{2})\right) \mathrm{d}u \\
&= \left(\beta + \frac{\Delta^2}{2}\right)^{-\frac{n}{2}-\alpha} \Gamma(\alpha + \frac{n}{2}).
\end{aligned}$$

Arranging terms and re-introducing the constant factor eventually yields the result of (A.26a):

$$\begin{aligned}
p_x(x) &= \frac{\Gamma(\alpha + \frac{n}{2})}{\Gamma(\alpha)} \frac{\beta^\alpha}{(2\pi)^{\frac{n}{2}}} \frac{1}{\sqrt{\det(P)}} \left(\beta + \frac{\Delta^2}{2}\right)^{-\frac{n}{2}-\alpha} \\
&= \frac{\Gamma(\alpha + \frac{n}{2})}{\Gamma(\alpha)} \frac{1}{(2\beta\pi)^{\frac{n}{2}}} \frac{1}{\sqrt{\det(P)}} \left(1 + \frac{\Delta^2}{2\beta}\right)^{-\frac{n}{2}-\alpha}.
\end{aligned} \tag{A.40}$$

Although not visible at first sight, the parameters β and P are related. To show this, we investigate the parts of (A.40) where β and P occur,

$$\begin{aligned}
\frac{\Delta^2}{\beta} &= (x - \hat{x})^{\mathrm{T}} (\beta P)^{-1} (x - \hat{x}), \\
\sqrt{\det(P)} \beta^{\frac{n}{2}} &= \sqrt{\det(\beta P)},
\end{aligned} \tag{A.41}$$

and see that the product of both determines the density. This means that any random variable generated using (A.24) with $v \sim \mathrm{Gam}(\alpha, \beta/c)$ and $y \sim \mathcal{N}(y; 0, cP)$ admits the same distribution for any $c > 0$. Consequently, we can choose $c = \beta/\alpha$ and thereby have $v \sim \mathrm{Gam}(\alpha, \alpha)$ without loss of generality. More specifically, we can replace $\alpha = \beta = \nu/2$ in (A.40) and recover the "common" t density of (A.23).

A.3.7 Useful Matrix Computations

We here present results that are used in the derivation of the conditional density of a partitioned t random vector, as presented in Appendix A.3.8. However, similar expressions show up when working with the Gaussian distribution.

We consider a partitioned random vector x that admits the density (A.23) with

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \hat{x} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}, \quad P = \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^{\mathrm{T}} & P_{22} \end{bmatrix}, \tag{A.42}$$

where x_1 and x_2 have dimension n_1 and n_2, respectively. Mean and matrix parameter are composed of blocks of appropriate dimensions.

Sometimes it is more convenient to work with the inverse of the matrix P, which is another block matrix:

$$\begin{bmatrix} P_{11} & P_{12} \\ P_{12}^{\mathrm{T}} & P_{22} \end{bmatrix}^{-1} = \begin{bmatrix} L_{11} & L_{12} \\ L_{12}^{\mathrm{T}} & L_{22} \end{bmatrix}. \tag{A.43}$$

Using formulas for block matrix inversion Kailath et al. (2000) we can find the expressions

$$\begin{aligned} L_{11} &= (P_{11} - P_{12}P_{22}^{-1}P_{12}^{\mathrm{T}})^{-1}, && \text{(A.44a)} \\ L_{12} &= -(P_{11} - P_{12}P_{22}^{-1}P_{12}^{\mathrm{T}})^{-1}P_{12}P_{22}^{-1} \\ &= -L_{11}P_{12}P_{22}^{-1}, && \text{(A.44b)} \\ P_{22} &= (L_{22} - L_{12}^{\mathrm{T}}L_{11}^{-1}L_{12})^{-1}. && \text{(A.44c)} \end{aligned}$$

Furthermore, we have for the determinant of P

$$\begin{aligned} \det(\begin{bmatrix} P_{11} & P_{12} \\ P_{12}^{\mathrm{T}} & P_{22} \end{bmatrix}) &= \det(P_{22}) \det(P_{11} - P_{12}P_{22}^{-1}P_{12}^{\mathrm{T}}) \\ &= \frac{\det(P_{22})}{\det(L_{11})}. \end{aligned} \tag{A.45}$$

We next investigate the quadratic form Δ^2 in which x enters the t density (A.23). Inserting the parameter blocks (A.42) and grouping powers of x_1 yields

$$\begin{aligned} \Delta^2 &= (x - \hat{x})^{\mathrm{T}} L (x - \hat{x}) \\ &= x_1^{\mathrm{T}} L_{11} x_1 - 2x_1^{\mathrm{T}}(L_{11}\hat{x}_1 - L_{12}(x_2 - \hat{x}_2)) \\ &\quad + \hat{x}_1^{\mathrm{T}} L_{11}\hat{x}_1 - 2\hat{x}_1^{\mathrm{T}} L_{12}(x_2 - \hat{x}_2) + (x_2 - \hat{x}_2)^{\mathrm{T}} L_{22}(x_2 - \hat{x}_2). \end{aligned}$$

With the intention of establishing compact quadratic forms we introduce the following quantities:

$$\begin{aligned} \hat{x}_{1|2} &= (L_{11})^{-1}(L_{11}\hat{x}_1 - L_{12}(x_2 - \hat{x}_2)) \\ &= \hat{x}_1 - L_{11}^{-1}L_{12}(x_2 - \hat{x}_2), && \text{(A.46a)} \\ \Delta_{1*}^2 &= (x_1 - \hat{x}_{1|2})^{\mathrm{T}} L_{11}(x_1 - \hat{x}_{1|2}), && \text{(A.46b)} \end{aligned}$$

Completing the square in x_1 and subsequent simplification then yields

$$\begin{aligned} \Delta^2 &= \Delta_{1*}^2 - \hat{x}_{1|2}^{\mathrm{T}} L_{11}\hat{x}_{1|2} \\ &\quad + \hat{x}_1^{\mathrm{T}} L_{11}\hat{x}_1 - 2\hat{x}_1^{\mathrm{T}} L_{12}(x_2 - \hat{x}_2) + (x_2 - \hat{x}_2)^{\mathrm{T}} L_{22}(x_2 - \hat{x}_2) \\ &= \Delta_{1*}^2 + (x_2 - \hat{x}_2)^{\mathrm{T}} \Big(L_{22} - L_{12}^{\mathrm{T}} L_{11}^{-1} L_{12}\Big)(x_2 - \hat{x}_2) \\ &= \Delta_{1*}^2 + \Delta_2^2, \end{aligned} \tag{A.47}$$

where we used (A.44c) and introduced $\Delta_2^2 = (x_2 - \hat{x}_2)^{\mathrm{T}} P_{22}^{-1}(x_2 - \hat{x}_2)$.

A.3.8 Derivation of the Conditional *t* Density

We here derive the conditional density of $x_1|x_2$, as presented in Section A.3.4. Although basic probability theory provides the result as the ratio of the joint and the marginal density, the expressions take a very complicated form. Step by step, we disclose that the resulting density is again Student's t. In the course of the derivations, we make heavy use of the results of Appendix A.3.7. We furthermore work on the α, β parametrization of the t density given in (A.26a).

The conditional density is given as ratio of joint (x_1 and x_2) and marginal (x_2 only) density:

$$\begin{aligned} p(x_1|x_2) &= \frac{p(x_1,x_2)}{p(x_2)} \\ &= \frac{\Gamma(\alpha+\frac{n}{2})}{\Gamma(\alpha)} \frac{1}{(2\beta\pi)^{\frac{n}{2}}} \frac{1}{\sqrt{\det(P)}} \left(1+\frac{\Delta^2}{2\beta}\right)^{-\frac{n}{2}-\alpha} \\ &\left(\frac{\Gamma(\alpha+\frac{n_2}{2})}{\Gamma(\alpha)} \frac{1}{(2\beta\pi)^{\frac{n_2}{2}}} \frac{1}{\sqrt{\det(P_{22})}} \left(1+\frac{\Delta_2^2}{2\beta}\right)^{-\frac{n_2}{2}-\alpha} \right)^{-1} . \end{aligned} \tag{A.48}$$

Here, we combined (A.28), (A.29), and (A.26a).

For the moment, we ignore the normalization constants and focus on the bracket term of $p(x_1, x_2)$, which is colored in red in (A.48). Using the result (A.47) we can rewrite it:

$$\begin{aligned} \left(1+\frac{\Delta^2}{2\beta}\right)^{-\frac{n}{2}-\alpha} &= \left(1+\frac{\Delta_2^2}{2\beta}+\frac{\Delta_{1*}^2}{2\beta}\right)^{-\frac{n}{2}-\alpha} \\ &= \left(1+\frac{\Delta_2^2}{2\beta}\right)^{-\frac{n}{2}-\alpha} \left(1+\frac{\Delta_{1*}^2}{2\beta+\Delta_2^2}\right)^{-\frac{n}{2}-\alpha} . \end{aligned}$$

The bracket expressions (both red and blue) in (A.48) now simplify to

$$\begin{aligned} &\left(1+\frac{\Delta_2^2}{2\beta}\right)^{-\frac{n}{2}-\alpha} \left(1+\frac{\Delta_{1*}^2}{2\beta+\Delta_2^2}\right)^{-\frac{n}{2}-\alpha} \left(1+\frac{\Delta_2^2}{2\beta}\right)^{\frac{n_2}{2}+\alpha} \\ &= \left(1+\frac{\Delta_2^2}{2\beta}\right)^{-\frac{n_1}{2}} \left(1+\frac{\Delta_{1*}^2}{2\beta+\Delta_2^2}\right)^{-\frac{n}{2}-\alpha} . \end{aligned} \tag{A.49}$$

The remaining bits of (A.48), that is the ratio of normalization constants, can be written as

$$\frac{\Gamma(\alpha+\frac{n}{2})}{\Gamma(\alpha+\frac{n_2}{2})} \frac{(2\beta\pi)^{\frac{n_2}{2}}}{(2\beta\pi)^{\frac{n}{2}}} \frac{\sqrt{\det(P_{22})}}{\sqrt{\det(P)}} = \frac{\Gamma(\alpha+\frac{n_2}{2}+\frac{n_1}{2})}{\Gamma(\alpha+\frac{n_2}{2})} \frac{1}{(2\beta\pi)^{\frac{n_1}{2}}} \sqrt{\det(L_{11})}, \tag{A.50}$$

where we used the determinant result (A.45).

Combining the above results (A.49) and (A.50) gives the (still complicated) expression

$$p(x_1|x_2) = \frac{\Gamma(\alpha + \frac{n_2}{2} + \frac{n_1}{2})}{\Gamma(\alpha + \frac{n_2}{2})} \frac{\sqrt{\det(L_{11})}}{(2\beta\pi)^{\frac{n_1}{2}}} \left(1 + \frac{\Delta_2^2}{2\beta}\right)^{-\frac{n_1}{2}} \left(1 + \frac{\Delta_{1*}^2}{2\beta + \Delta_2^2}\right)^{-\frac{n}{2}-\alpha},$$

which we shall now slowly shape into a recognizable t density.

Recalling from (A.46b) that $\Delta_{1*}^2 = (x_1 - \hat{x}_{1|2})^{\mathrm{T}} L_{11} (x_1 - \hat{x}_{1|2})$, and introducing $\alpha_{1|2} = \alpha + \frac{n_2}{2}$, yields

$$\begin{aligned} p(x_1|x_2) &= \frac{\Gamma(\alpha_{1|2} + \frac{n_1}{2})}{\Gamma(\alpha_{1|2})} \frac{\sqrt{\det(L_{11})}}{(2\beta\pi)^{\frac{n_1}{2}}} \left(1 + \frac{\Delta_2^2}{2\beta}\right)^{-\frac{n_1}{2}} \left(1 + \frac{\Delta_{1*}^2}{2\beta + \Delta_2^2}\right)^{-\frac{n_1}{2}-\alpha_{1|2}} \\ &= \frac{\Gamma(\alpha_{1|2} + \frac{n_1}{2})}{\Gamma(\alpha_{1|2})} \frac{\sqrt{\det(L_{11})}}{\left(\pi\left(2\beta + \Delta_2^2\right)\right)^{\frac{n_1}{2}}} \left(1 + \frac{(x_1 - \hat{x}_{1|2})^{\mathrm{T}} L_{11} (x_1 - \hat{x}_{1|2})}{2\beta + \Delta_2^2}\right)^{-\frac{n_1}{2}-\alpha_{1|2}}. \end{aligned}$$

Comparison with (A.26b) reveals that this is also a t distribution, but parametrized in terms of α and β, with

$$\begin{aligned} \beta_{1|2}^* &= \frac{1}{2}(2\beta + \Delta_2^2) = \beta + \frac{1}{2}(x_2 - \hat{x}_2)^{\mathrm{T}} P_{22}^{-1} (x_2 - \hat{x}_2), \\ P_{1|2}^* &= L_{11}^{-1} = P_{11} - P_{12} P_{22}^{-1} P_{12}^{\mathrm{T}}. \end{aligned}$$

Obviously, $\alpha_{1|2} \neq \beta_{1|2}^*$. Because, however, only the product $\beta_{1|2}^* P_{1|2}^*$ enters a t density, the parameters can be adjusted as discussed in Appendix A.3.2 and Appendix A.3.6. Converting back to ν parametrization, as in (A.23), we obtain $p(x_1|x_2) = \mathrm{St}(x_1; \hat{x}_{1|2}, P_{1|2}, \nu_{1|2})$ with

$$\begin{aligned} \nu_{1|2} &= 2\alpha_{1|2} = 2\alpha + n_2 = \nu + n_2, \\ \hat{x}_{1|2} &= \hat{x}_1 - L_{11}^{-1} L_{12} (x_2 - \hat{x}_2) \\ &= \hat{x}_1 + P_{12} P_{22}^{-1} (x_2 - \hat{x}_2), \\ P_{1|2} &= \frac{\beta_{1|2}^*}{\alpha_{1|2}} P_{1|2}^* \\ &= \frac{\beta + \frac{1}{2}(x_2 - \hat{x}_2)^{\mathrm{T}} P_{22}^{-1} (x_2 - \hat{x}_2)}{\alpha + \frac{n_2}{2}} (P_{11} - P_{12} P_{22}^{-1} P_{12}^{\mathrm{T}}) \\ &= \frac{\nu + (x_2 - \hat{x}_2)^{\mathrm{T}} P_{22}^{-1} (x_2 - \hat{x}_2)}{\nu + n_2} (P_{11} - P_{12} P_{22}^{-1} P_{12}^{\mathrm{T}}), \end{aligned}$$

which are the parameters provided in Section A.3.4.

Bibliography

M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*, volume 55. Dover publications, 1965.

G. Agamennoni, J. I. Nieto, and E. M. Nebot. Approximate inference in state-space models with heavy-tailed noise. *IEEE Transactions on Signal Processing*, 60(10):5024 –5037, Oct. 2012.

B. D. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, June 1979.

T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley-Interscience, third edition, 2003.

I. Arasaratnam and S. Haykin. Cubature Kalman filters. *IEEE Transactions on Automatic Control*, 54(6):1254–1269, June 2009.

M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, Feb. 2002.

M. Athans, R. Wishner, and A. Bertolini. Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements. *IEEE Transactions on Automatic Control*, 13(5):504–514, 1968.

Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience, June 2001.

M. Baum, B. Noack, F. Beutler, D. Itte, and U. Hanebeck. Optimal Gaussian filtering for polynomial systems applied to association-free multi-target tracking. In *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, July 2011.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, Aug. 2006.

S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Aug. 1999.

O. Cappé, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, May 2007.

P. Closas, C. Fernandez-Prades, and J. Vila-Valls. Multiple quadrature Kalman filtering. *IEEE Transactions on Signal Processing*, 60(12):6125–6137, 2012.

M. H. DeGroot. *Optimal Statistical Decisions*. Wiley-Interscience, WCL edition, Apr. 2004.

M. Evans and T. Swartz. Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems. *Statistical Science*, 10(3): 254–272, Aug. 1995.

A. Gelb. *Applied Optimal Estimation*. MIT Press, May 1974.

F. Girón and J. Rojano. Bayesian Kalman filtering with elliptically contoured errors. *Biometrika*, 81(2):390–395, June 1994.

N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, Apr. 1993.

M. Grewal and A. Andrews. Applications of Kalman filtering in aerospace 1960 to the present. *IEEE Control Systems Magazine*, 30(3):69–78, 2010.

A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Society for Industrial and Applied Mathematics, second edition, Nov. 2008.

F. Gustafsson. Particle filter theory and practice with positioning applications. *Aerospace and Electronic Systems Magazine, IEEE*, 25(7):53–82, 2010a.

F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur AB, Mar. 2010b.

F. Gustafsson and G. Hendeby. Some relations between extended and unscented Kalman filters. *IEEE Transactions on Signal Processing*, 60(2):545 –555, Feb. 2012.

F. Gustafsson and A. Isaksson. Best choice of coordinate system for tracking coordinated turns. In *Proceedings of the 35th IEEE Conference on Decision and Control, 1996*, volume 3, pages 3145 –3150, Dec. 1996.

A. Gut. *An Intermediate Course in Probability*. Springer, second edition, June 2009.

J. E. Handschin and D. Q. Mayne. Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control*, 9(5):547–559, 1969.

U. Hanebeck, M. Huber, and V. Klumpp. Dirac mixture approximation of multivariate Gaussian densities. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 3851–3858, 2009.

G. Hendeby. *Performance and Implementaion Aspects of Nonlinear Filtering*. PhD thesis, Linköping University, Department of Electrical Engineering, Linköping, Sweden, Feb. 2008. Linköping Studies in Science and Technology. Dissertations. No. 1161.

G. Hendeby, R. Karlsson, and F. Gustafsson. The Rao-Blackwellized particle filter: A filter bank implementation. *EURASIP Journal on Advances in Signal Processing*, 2010(1), Dec. 2010.

K. Ito and K. Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45(5):910–927, May 2000.

A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, Mar. 1970.

B. Jia, M. Xin, and Y. Cheng. High-degree cubature Kalman filter. *Automatica*, 49 (2):510–518, Feb. 2013.

S. Julier. The scaled unscented transformation. In *Proceedings of the American Control Conference 2002*, volume 6, pages 4555–4559, 2002.

S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference 1995*, volume 3, pages 1628–1632, 1995.

S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477 –482, Mar. 2000.

S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401– 422, Mar. 2004.

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, Apr. 2000.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, Mar. 1960.

R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of basic Engineering*, 83(3):95–108, 1961.

G. Koop. *Bayesian Econometrics*. Wiley-Interscience, July 2003.

S. Kotz and S. Nadarajah. *Multivariate t Distributions and Their Applications*. Cambridge University Press, Feb. 2004.

T. Lefebvre, H. Bruyninckx, and J. De Schutter. *Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*. Springer, Oct. 2005.

F. Lindsten and T. Schön. Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends in Machine Learning*, 2013.

M. Luca, S. Azou, G. Burel, and A. Serbanescu. On exact Kalman filtering of polynomial systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53(6):1329 –1340, June 2006.

M. Mallick, M. Morelande, L. Mihaylova, S. Arulampalam, and Y. Yan. Angle-only filtering in three dimensions. In M. Mallick, V. Krishnamurthy, and B.-N. Vo, editors, *Integrated Tracking, Classification, and Sensor Management: Theory and Applications*, pages 3–42. John Wiley and Sons, 2012.

D. Manolakis. Kronecker product based second order approximation of mean value and covariance matrix in nonlinear transformations. *IEEE Signal Processing Letters*, 18(1):43 –46, Jan. 2011.

J. Mattingley and S. Boyd. Real-time convex optimization in signal processing. *IEEE Signal Processing Magazine*, 27(3):50 –61, May 2010.

P. S. Maybeck. *Stochastic Models, Estimation, and Control: Volume 1*. Academic Press, June 1979.

P. S. Maybeck. *Stochastic Models, Estimation, and Control: Volume 2*. Academic Press, June 1982.

G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley-Interscience, second edition, Mar. 2008.

R. J. Meinhold and N. D. Singpurwalla. Robustification of Kalman filter models. *Journal of the American Statistical Association*, 84(406):479–486, June 1989.

J. Nocedal and S. Wright. *Numerical Optimization*. Springer, second edition, July 2006.

M. Nørgaard, N. K. Poulsen, and O. Ravn. Advances in derivative-free state estimation for nonlinear systems. Technical Report IMM-REP-1998-15, Department of Mathematical Modelling, DTU, Lyngby, Denmark, Apr. 2000a.

M. Nørgaard, N. K. Poulsen, and O. Ravn. New developments in state estimation for nonlinear systems. *Automatica*, 36(11):1627–1638, Nov. 2000b.

A. Papoulis and S. U. Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Europe, fourth edition, Jan. 2002.

R. Piché, S. Särkkä, and J. Hartikainen. Recursive outlier-robust filtering and smoothing for nonlinear systems using the multivariate Student-t distribution. In *Proceedings of MLSP*, Sept. 2012.

C. R. Rao. *Linear Statistical Inference and its Applications*. Wiley-Interscience, second edition, Dec. 2001.

H. E. Rauch, C. T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, Aug. 1965.

B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, Jan. 2004.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer, second edition, 2004.

M. Roth. On the multivariate t distribution. Technical Report LiTH-ISY-R-3059, Automatic Control, Linköping University, Linköping, Sweden, Apr. 2013.

M. Roth and F. Gustafsson. An efficient implementation of the second order extended Kalman filter. In *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, July 2011.

M. Roth, F. Gustafsson, and U. Orguner. On-road trajectory generation from GPS data: A particle filtering/smoothing application. In *15th International Conference on Information Fusion (FUSION)*, pages 779 –786, July 2012.

M. Roth, E. Özkan, and F. Gustafsson. A Student's t filter for heavy tailed process and measurement noise. In *38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

W. J. Rugh. *Linear System Theory*. Prentice Hall, second edition, Aug. 1995.

S. Saha, P. K. Mandal, Y. Boers, H. Driessen, and A. Bagchi. Gaussian proposal density using moment matching in SMC methods. *Statistics and Computing*, 19 (2):203–208, June 2009.

J. Sarmavuori and S. Särkkä. Fourier-Hermite Kalman filter. *IEEE Transactions on Automatic Control*, 57(6):1511–1515, 2012.

T. S. Schei. A finite-difference method for linearization in nonlinear estimation algorithms. *Automatica*, 33(11):2053–2058, Nov. 1997.

M. Šimandl and J. Duník. Derivative-free estimation methods: New results and performance analysis. *Automatica*, 45(7):1749–1757, July 2009.

H. W. Sorenson. On the development of practical nonlinear filters. *Information Sciences*, 7:253–270, 1974.

H. W. Sorenson and A. R. Stubberud. Recursive filtering for systems with small but non-negligible non-linearities. *International Journal of Control*, 7(3):271–280, 1968.

A. H. Stroud. Some fifth degree integration formulas for symmetric regions II. *Numerische Mathematik*, 9(5):460–468, Apr. 1967.

A. H. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice-Hall, 1971.

A. H. Stroud and D. Secrest. Approximate integration formulas for certain spherically symmetric regions. *Mathematics of Computation*, 17(82):105–135, Apr. 1963.

P. Swerling. First-order error propagation in a stagewise smoothing procedure for satellite observations. Technical Report RM-2329, June 1959.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, Mass., 2005.

R. van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, OGI School of Science & Engineering, Oregon Health & Science University, Portland, OR, USA, Apr. 2004.

C. F. Van Loan. The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, 123(1–2):85–100, Nov. 2000.

M. Verhaegen and V. Verdult. *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, May 2007.

Y. Wu, D. Hu, M. Wu, and X. Hu. A numerical-integration perspective on Gaussian filters. *IEEE Transactions on Signal Processing*, 54(8):2910–2921, Aug. 2006.

Licentiate Theses
Division of Automatic Control
Linköping University

P. Andersson: Adaptive Forgetting through Multiple Models and Adaptive Control of Car Dynamics. Thesis No. 15, 1983.

B. Wahlberg: On Model Simplification in System Identification. Thesis No. 47, 1985.

A. Isaksson: Identification of Time Varying Systems and Applications of System Identification to Signal Processing. Thesis No. 75, 1986.

G. Malmberg: A Study of Adaptive Control Missiles. Thesis No. 76, 1986.

S. Gunnarsson: On the Mean Square Error of Transfer Function Estimates with Applications to Control. Thesis No. 90, 1986.

M. Viberg: On the Adaptive Array Problem. Thesis No. 117, 1987.

K. Ståhl: On the Frequency Domain Analysis of Nonlinear Systems. Thesis No. 137, 1988.

A. Skeppstedt: Construction of Composite Models from Large Data-Sets. Thesis No. 149, 1988.

P. A. J. Nagy: MaMiS: A Programming Environment for Numeric/Symbolic Data Processing. Thesis No. 153, 1988.

K. Forsman: Applications of Constructive Algebra to Control Problems. Thesis No. 231, 1990.

I. Klein: Planning for a Class of Sequential Control Problems. Thesis No. 234, 1990.

F. Gustafsson: Optimal Segmentation of Linear Regression Parameters. Thesis No. 246, 1990.

H. Hjalmarsson: On Estimation of Model Quality in System Identification. Thesis No. 251, 1990.

S. Andersson: Sensor Array Processing; Application to Mobile Communication Systems and Dimension Reduction. Thesis No. 255, 1990.

K. Wang Chen: Observability and Invertibility of Nonlinear Systems: A Differential Algebraic Approach. Thesis No. 282, 1991.

J. Sjöberg: Regularization Issues in Neural Network Models of Dynamical Systems. Thesis No. 366, 1993.

P. Pucar: Segmentation of Laser Range Radar Images Using Hidden Markov Field Models. Thesis No. 403, 1993.

H. Fortell: Volterra and Algebraic Approaches to the Zero Dynamics. Thesis No. 438, 1994.

T. McKelvey: On State-Space Models in System Identification. Thesis No. 447, 1994.

T. Andersson: Concepts and Algorithms for Non-Linear System Identifiability. Thesis No. 448, 1994.

P. Lindskog: Algorithms and Tools for System Identification Using Prior Knowledge. Thesis No. 456, 1994.

J. Plantin: Algebraic Methods for Verification and Control of Discrete Event Dynamic Systems. Thesis No. 501, 1995.

J. Gunnarsson: On Modeling of Discrete Event Dynamic Systems, Using Symbolic Algebraic Methods. Thesis No. 502, 1995.

A. Ericsson: Fast Power Control to Counteract Rayleigh Fading in Cellular Radio Systems. Thesis No. 527, 1995.

M. Jirstrand: Algebraic Methods for Modeling and Design in Control. Thesis No. 540, 1996.

K. Edström: Simulation of Mode Switching Systems Using Switched Bond Graphs. Thesis No. 586, 1996.

J. Palmqvist: On Integrity Monitoring of Integrated Navigation Systems. Thesis No. 600, 1997.

A. Stenman: Just-in-Time Models with Applications to Dynamical Systems. Thesis No. 601, 1997.

M. Andersson: Experimental Design and Updating of Finite Element Models. Thesis No. 611, 1997.

U. Forssell: Properties and Usage of Closed-Loop Identification Methods. Thesis No. 641, 1997.

M. Larsson: On Modeling and Diagnosis of Discrete Event Dynamic systems. Thesis No. 648, 1997.

N. Bergman: Bayesian Inference in Terrain Navigation. Thesis No. 649, 1997.

V. Einarsson: On Verification of Switched Systems Using Abstractions. Thesis No. 705, 1998.

J. Blom, F. Gunnarsson: Power Control in Cellular Radio Systems. Thesis No. 706, 1998.

P. Spångéus: Hybrid Control using LP and LMI methods – Some Applications. Thesis No. 724, 1998.

M. Norrlöf: On Analysis and Implementation of Iterative Learning Control. Thesis No. 727, 1998.

A. Hagenblad: Aspects of the Identification of Wiener Models. Thesis No. 793, 1999.

F. Tjärnström: Quality Estimation of Approximate Models. Thesis No. 810, 2000.

C. Carlsson: Vehicle Size and Orientation Estimation Using Geometric Fitting. Thesis No. 840, 2000.

J. Löfberg: Linear Model Predictive Control: Stability and Robustness. Thesis No. 866, 2001.

O. Härkegård: Flight Control Design Using Backstepping. Thesis No. 875, 2001.

J. Elbornsson: Equalization of Distortion in A/D Converters. Thesis No. 883, 2001.

J. Roll: Robust Verification and Identification of Piecewise Affine Systems. Thesis No. 899, 2001.

I. Lind: Regressor Selection in System Identification using ANOVA. Thesis No. 921, 2001.

R. Karlsson: Simulation Based Methods for Target Tracking. Thesis No. 930, 2002.

P.-J. Nordlund: Sequential Monte Carlo Filters and Integrated Navigation. Thesis No. 945, 2002.

M. Östring: Identification, Diagnosis, and Control of a Flexible Robot Arm. Thesis No. 948, 2002.

C. Olsson: Active Engine Vibration Isolation using Feedback Control. Thesis No. 968, 2002.

J. Jansson: Tracking and Decision Making for Automotive Collision Avoidance. Thesis No. 965, 2002.

N. Persson: Event Based Sampling with Application to Spectral Estimation. Thesis No. 981, 2002.

D. Lindgren: Subspace Selection Techniques for Classification Problems. Thesis No. 995, 2002.

E. Geijer Lundin: Uplink Load in CDMA Cellular Systems. Thesis No. 1045, 2003.

M. Enqvist: Some Results on Linear Models of Nonlinear Systems. Thesis No. 1046, 2003.

T. Schön: On Computational Methods for Nonlinear Estimation. Thesis No. 1047, 2003.

F. Gunnarsson: On Modeling and Control of Network Queue Dynamics. Thesis No. 1048, 2003.

S. Björklund: A Survey and Comparison of Time-Delay Estimation Methods in Linear Systems. Thesis No. 1061, 2003.

M. Gerdin: Parameter Estimation in Linear Descriptor Systems. Thesis No. 1085, 2004.

A. Eidehall: An Automotive Lane Guidance System. Thesis No. 1122, 2004.

E. Wernholt: On Multivariable and Nonlinear Identification of Industrial Robots. Thesis No. 1131, 2004.

J. Gillberg: Methods for Frequency Domain Estimation of Continuous-Time Models. Thesis No. 1133, 2004.

G. Hendeby: Fundamental Estimation and Detection Limits in Linear Non-Gaussian Systems. Thesis No. 1199, 2005.

D. Axehill: Applications of Integer Quadratic Programming in Control and Communication. Thesis No. 1218, 2005.

J. Sjöberg: Some Results On Optimal Control for Nonlinear Descriptor Systems. Thesis No. 1227, 2006.

D. Törnqvist: Statistical Fault Detection with Applications to IMU Disturbances. Thesis No. 1258, 2006.

H. Tidefelt: Structural algorithms and perturbations in differential-algebraic equations. Thesis No. 1318, 2007.

S. Moberg: On Modeling and Control of Flexible Manipulators. Thesis No. 1336, 2007.

J. Wallén: On Kinematic Modelling and Iterative Learning Control of Industrial Robots. Thesis No. 1343, 2008.

J. Harju Johansson: A Structure Utilizing Inexact Primal-Dual Interior-Point Method for Analysis of Linear Differential Inclusions. Thesis No. 1367, 2008.

J. D. Hol: Pose Estimation and Calibration Algorithms for Vision and Inertial Sensors. Thesis No. 1370, 2008.

H. Ohlsson: Regression on Manifolds with Implications for System Identification. Thesis No. 1382, 2008.

D. Ankelhed: On low order controller synthesis using rational constraints. Thesis No. 1398, 2009.

P. Skoglar: Planning Methods for Aerial Exploration and Ground Target Tracking. Thesis No. 1420, 2009.

C. Lundquist: Automotive Sensor Fusion for Situation Awareness. Thesis No. 1422, 2009.

C. Lyzell: Initialization Methods for System Identification. Thesis No. 1426, 2009.

R. Falkeborn: Structure exploitation in semidefinite programming for control. Thesis No. 1430, 2010.

D. Petersson: Nonlinear Optimization Approaches to $\mathcal{H}_2$-Norm Based LPV Modelling and Control. Thesis No. 1453, 2010.

Z. Sjanic: Navigation and SAR Auto-focusing in a Sensor Fusion Framework. Thesis No. 1464, 2011.

K. Granström: Loop detection and extended target tracking using laser data. Thesis No. 1465, 2011.

J. Callmer: Topics in Localization and Mapping. Thesis No. 1489, 2011.

F. Lindsten: Rao-Blackwellised particle methods for inference and identification. Thesis No. 1480, 2011.

M. Skoglund: Visual Inertial Navigation and Calibration. Thesis No. 1500, 2011.

S. Khoshfetrat Pakazad: Topics in Robustness Analysis. Thesis No. 1512, 2011.

P. Axelsson: On Sensor Fusion Applied to Industrial Manipulators. Thesis No. 1511, 2011.

A. Carvalho Bittencourt: On Modeling and Diagnosis of Friction and Wear in Industrial Robots. Thesis No. 1516, 2012.

P. Rosander: Averaging level control in the presence of frequent inlet flow upsets . Thesis No. 1527, 2012.

N. Wahlström: Localization using Magnetometers and Light Sensors. Thesis No. 1581, 2013.

R. Larsson: System Identification of Flight Mechanical Characteristics. Thesis No. 1599, 2013.

Y. Jung: Estimation of Inverse Models Applied to Power Amplifier Predistortion. Thesis No. 1605, 2013.

M. Syldatk: On Calibration of Ground Sensor Networks. Thesis No. 1611, 2013.