AALTO UNIVERSITY SCHOOL OF SCIENCE AND TECHNOLOGY
Faculty of Information and Natural Sciences

Arno Solin

CUBATURE INTEGRATION METHODS IN NON-LINEAR KALMAN FILTERING AND SMOOTHING

Bachelor's thesis

Espoo 2010

Thesis supervisor:

Prof. Harri Ehtamo

Thesis instructor:

D.Sc.(Tech.) Simo Särkkä

Aalto University

AALTO UNIVERSITY SCHOOL OF SCIENCE AND TECHNOLOGY PO Box 1100, FI-00076 AALTO http://www.aalto.fi	ABSTRACT OF THE BACHELOR'S THESIS

Author:	Arno Solin

Title:	Cubature Integration Methods in Non-Linear Kalman Filtering and Smoothing
Faculty:	Faculty of Information and Natural Sciences
Degree programme:	Engineering Physics and Mathematics

Major subject:	Systems analysis	Major subject code:	F3010

Supervisor:	Prof. Harri Ehtamo
Instructor:	D.Sc.(Tech.) Simo Särkkä

Optimal estimation problems arise in various different settings where indirect noisy observations are used to determine the underlying state of a time-varying system. For systems with non-linear dynamics there exist various methods that extend linear filtering and smoothing methods to handle non-linearities.

In this thesis the non-linear optimal estimation framework is presented with the help of an assumed density approach. The Gaussian integrals that arise in this setting are solved using two different cubature integration methods.

Cubature integration extends the weighted sum approach from univariate quadrature methods to multidimensional cubature methods. In this thesis the focus is put on two methods that use deterministically chosen sigma points to form the desired approximation. The Gauss–Hermite rule uses a simple product rule method to fill the multidimensional space with cubature points, whereas the spherical–radial rule uses invariant theory to diminish the number of points by utilizing symmetries.

The derivations of the Gauss–Hermite and spherical–radial rules are reviewed. The corresponding non-linear Kalman filter and Rauch–Tung–Striebel smoother algorithms are presented. Additionally, the relation between the cubature rules and the unscented transformation is discussed. It is also shown that the cubature Kalman filter can be interpreted as a refinement of the unscented Kalman filter.

Date: September 12, 2010	Language: English	Number of pages: 40 + 5

Keywords: Kalman filter, assumed density filter, cubature integration

Preface

This work was carried out in the Bayesian Statistical Methods group in the Centre of Excellence in Computational Complex Systems Research at Aalto University, Finland.

I would like express my gratitude to my instructor Dr. Simo Särkkä for providing me a constant flow of good advice, relevant books and helpful thoughts. Additionally, I would like to thank prof. Harri Ehtamo for supervising this thesis, M.Sc. Jouni Hartikainen for codes and Dr. Aki Vehtari for employing me.

Otaniemi, 2010
Arno Solin

Contents

Symbols and Abbreviations

Matrices are capitalized and vectors are in bold type. We do not generally distinguish between probabilities and probability densities.

Operators and miscellaneous notation

$1:k$	$1, 2, \ldots, k$
$p(\mathbf{x} \mid \mathbf{y})$	Conditional probability density of $\mathbf{x}$ given $\mathbf{y}$
$\mathbf{x}_{k\mid k-1}$	Conditional value of $\mathbf{x}_k$ given values up to step $k-1$
$\mathbb{R}$	The real numbers
$\mathbb{R}_+$	The positive real numbers
$\Gamma(\cdot)$	The gamma function
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\mathcal{U}(a, b)$	Uniform distribution between a and b
$\mathbf{I}$	Identity matrix
$\mathbf{A}^\mathsf{T}$	Matrix transpose
$\lvert\mathbf{A}\rvert$	Matrix determinant of $\mathbf{A}$
$\mathrm{chol}(\mathbf{A})$	Cholesky decomposition: $\mathrm{chol}(\mathbf{A}) = \mathbf{L}$, $\mathbf{L}\mathbf{L}^\mathsf{T} = \mathbf{A}$
$\mathrm{diag}(\mathbf{a})$	A diagonal matrix with elements of $\mathbf{a}$ on its diagonal

General notation

$\mathbf{x}$	System state
$\mathbf{y}$	Observation
k	Time step
T	Final time step
$\mathbf{q}_k$	Zero-mean (Gaussian) Process noise
$\mathbf{r}_k$	Zero-mean (Gaussian) Measurement noise
$\mathbf{Q}_k$	Process noise covariance
$\mathbf{R}_k$	Measurement noise covariance

Abbreviations

RTS	Rauch–Tung–Striebel smoother
EKF	Extended Kalman filter
ERTS	Extended Rauch–Tung–Striebel smoother
UKF	Unscented Kalman filter
URTS	Unscented Rauch–Tung–Striebel smoother
GHKF	Gauss–Hermite Kalman filter
GHRTS	Gauss–Hermite Rauch–Tung–Striebel smoother
CKF	Cubature Kalman filter
CRTS	Cubature Rauch–Tung–Striebel smoother

1 Introduction

The term *optimal estimation* refers to the methods used to estimate the underlying state of a time-varying system of which there exist only indirectly observed noisy measurements. These kinds of models can be found, for example, in navigation, target tracking, biological processes, telecommunications, audio signal processing, stochastic optimal control, physical processes, finance and learning systems. (see, *e.g.*, Särkkä, 2006)

The celebrated Kalman filter (Kalman, 1960) provides a closed-form solution to estimate the states of phenomena with linear underlying dynamics and linear observation models. Similarly the Rauch–Tung–Striebel smoother (Rauch *et al.*, 1965) provides the linear smoothing estimate. Non-linear dynamics and observations require approximating as the non-linearities do not preserve the Gaussian nature as such.

The extended Kalman filter (EKF) (Jazwinski, 1970) has been the *de facto* method for non-linear Kalman filtering for decades. Even so, the EKF has many undesired limitations. The linearization of the dynamical and measurement models rely on their first order derivatives, which requires the models to be continuous and differentiable and causes the Gaussian approximation to be local. As a consequence the extended Kalman filter does not work on considerable non-linearities.

Many refinements to the non-linear Gaussian approximation based filters have been presented throughout the years. Of these methods in the scope of this thesis are filter and smoother formulations that fall under the category of assumed density and sigma point filters. An exemplar of this category is the well-known unscented Kalman filter (UKF) (Julier and Uhlmann, 1996).

In the first section of this thesis we will go through the formulation of the linear Kalman filter and the Rauch–Tung–Striebel smoother and how this is extended to the non-linear assumed density form. In the context of assumed density filtering the Gaussian approximation falls back on solving a set of Gaussian integrals.

In the second section, we study two different cubature integration methods for solving the integrals. The first method extends the one-dimensional Gauss–Hermite quadrature method to a multidimensional cubature method. The second approach follows the derivation of Arasaratnam and Haykin (2009) to form a spherical–radial rule based cubature method.

Combining the two cubature integration methods with the assumed density filter and smoother formulation yields two different optimal estimation tools: The algorithms for the Gauss–Hermite Kalman filter (GHKF) and the Gauss–Hermite Rauch–Tung–Striebel smoother (GHRTS) together with the algorithms for the spherical–radial rule based cubature Kalman filter (CKF) and cubature Rauch–Tung–Striebel smoother (CRTS) are presented in the third section.

Two case studies are presented for demonstrating the characteristics of the estimation methods. They employ the methods on a target tracking problem and a neural network problem with an excessive number of dimensions.

The main contributions of this thesis are to (i) verify previous work done on the field, (ii) unify the formulation of different cubature and sigma point filter approaches, and (iii) discuss the properties of the cubature Kalman filter as a refinement of the unscented Kalman filter.

2 Gaussian Approximation Based Estimation

2.1 Linear Estimation

We start by considering a linear stochastic state-space model of the form

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{q}_{k-1} \\ \mathbf{y}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_k, \end{aligned} \tag{1}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the state, $\mathbf{y}_k \in \mathbb{R}^m$ is the measurement of the state at time step k, $\mathbf{q}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ is the *i.i.d.* Gaussian process noise of the dynamic model, $\mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is the *i.i.d.* Gaussian noise process of the measurement model, $\mathbf{A}_{k-1}$ is the dynamic model translation matrix and $\mathbf{H}_k$ is the measurement model matrix. (Kalman, 1960; Bar-Shalom *et al.*, 2001)

This is a special — but often encountered — case where the process and measurement noises are purely additive. The state-space model in (1) is said to be a *discrete-time* model because time is discretized. The time steps k run from 0 to T, and at time step $k = 0$ only the prior distribution is given, $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{m}_0, \mathbf{P}_0)$.

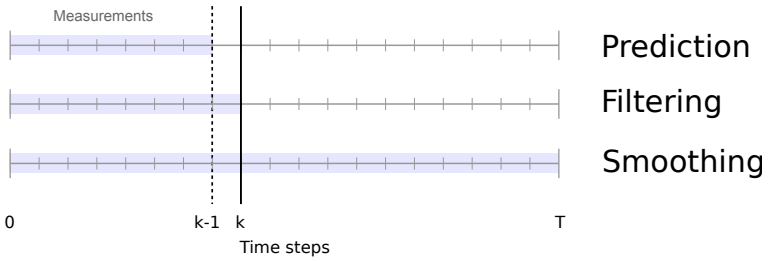


Figure 1: This diagram demonstrates the fundamental difference between *prediction*, *filtering* and (fixed interval) *smoothing*.

The dynamic model defines the system dynamics and its uncertainties as a *Markov sequence*. The dynamics are defined as transitions from the previous state with the accompanying uncertainty coming from the Gaussian term $\mathbf{q}_{k-1}$. The transitions may be modeled with the help of the transition distribution $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$. The measurement model maps the actual states $\mathbf{x}_k$ to measurements $\mathbf{y}_k$ which contain Gaussian noise. The dependency between the states and measurements can be shown in terms of a probability distribution $p(\mathbf{y}_k \mid \mathbf{x}_k)$.

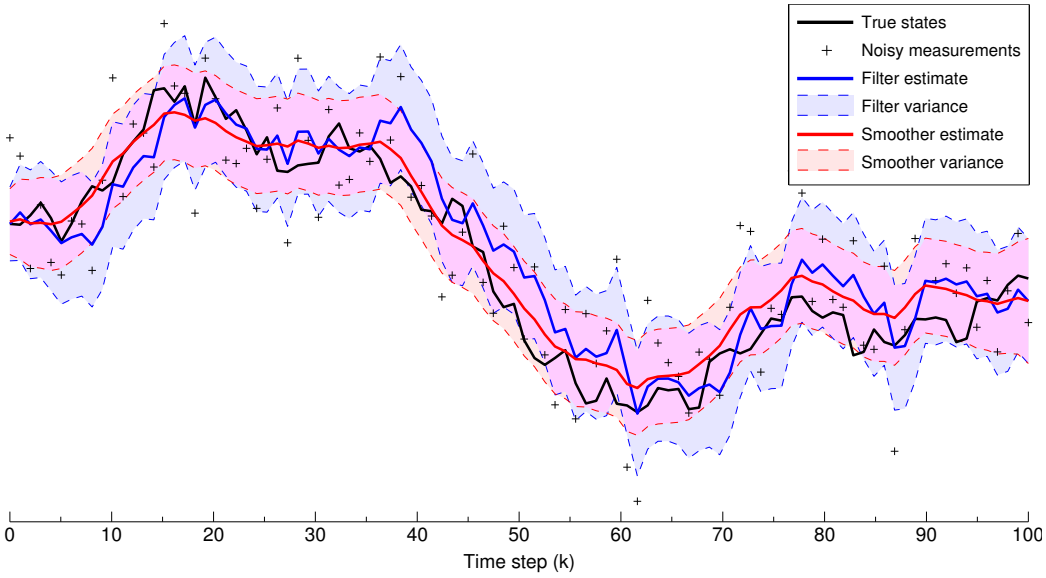


Figure 2: An illustrative example of the filtering and smoothing results for a linear Gaussian random-walk model. The variances are presented with the help of the 95 % confidence intervals.

The discrete-time state-space model presented in Equations (1) can be written equivalently in terms of probability distributions as a recursively defined probabilistic model of the form

$$\begin{aligned} p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) &= \mathcal{N}(\mathbf{x}_k \mid \mathbf{A}_{k-1}\mathbf{x}_{k-1}, \mathbf{Q}_{k-1}) \\ p(\mathbf{y}_k \mid \mathbf{x}_k) &= \mathcal{N}(\mathbf{y}_k \mid \mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k). \end{aligned} \tag{2}$$

The model is assumed to be Markovian in the sense that it incorporates the *Markov property*, which means that the current state is independent from the past prior to the previous state. Additionally all the measurements of the separate states are assumed to be conditionally independent of each other.

In this approach we bluntly divide the concept of Gaussian optimal estimation into three marginal distributions of interest (see, *e.g.*, Särkkä, 2006):

Filtering distributions $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$ that are the marginal distributions of current state $\mathbf{x}_k$ given all previous measurements $\mathbf{y}_{1:k} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k)$.

Prediction distributions $p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:k})$ that are the marginal distributions of forthcoming states.

Smoothing distributions $p(\mathbf{x}_k \mid \mathbf{y}_{1:T})$ that are the marginal distributions of the states $\mathbf{x}_k$ given measurements $\mathbf{y}_{1:T}$ such that $T > k$.

In Figure 1 the differences between optimal prediction, filtering and smoothing are demonstrated with the help of a time line of measurements. At time step k the prediction distribution utilizes less than k measurements, whereas the filtering solution uses exactly k measurements and the smoothing distribution more than k measurements.

An illustrative example of the differences between filtering and smoothing is shown in Figure 2. The black solid line in the figure demonstrates a realization of a Gaussian random walk process. The blue line together with the bluish patch following the line show the filtered solution obtained by using the noisy measurements in the figure. Similarly the red line and the reddish patch depict the smoothed solution. As the smoother has access to more measurements, it follows the original states more strictly and has a smaller variance than the filtering solution.

2.1.1 Kalman Filter Equations

The *Kalman filter* is a closed-form solution to the linear filtering problem in Equation (1) — or equivalently in (2). As the Kalman filter is conditional to all measurements up to time step k, the recursive filtering algorithm can be seen as a two-step process that first includes calculating the marginal distribution of the next step using the known system dynamics (see, *e.g.*, Bar-Shalom *et al.*, 2001). This is called the *Prediction step:*

$$\begin{aligned} \mathbf{m}_{k|k-1} &= \mathbf{A}_{k-1}\mathbf{m}_{k-1|k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^{\mathsf{T}} + \mathbf{Q}_{k-1}. \end{aligned} \tag{3}$$

The algorithm then uses the observation to update the distribution to match the new information obtained by the measurement at step k. This is called the *Update step*:

$$
\begin{aligned}
\mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\mathsf{T} + \mathbf{R}_k \\
\mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^\mathsf{T} \mathbf{S}_k^{-1} \\
\mathbf{m}_{k|k} &= \mathbf{m}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \mathbf{m}_{k|k-1}) \\
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^\mathsf{T}.
\end{aligned}
\tag{4}
$$

As a result the filtered distribution at step k is given by $\mathbf{x}_{k|k} \sim \mathcal{N}(\mathbf{m}_{k|k}, \mathbf{P}_{k|k})$. The difference $\mathbf{y}_k - \mathbf{H}_k \mathbf{m}_{k|k-1}$ in Equation (4) is called the innovation or the residual. It basically reflects the deflection between the actual measurement and the predicted measurement. The innovation is weighted by the *Kalman gain*. This term minimizes the *a posteriori* error covariance by weighting the residual with respect to the prediction step covariance $\mathbf{P}_{k|k-1}$ (see Maybeck, 1979; Welch and Bishop, 1995).

The linear Kalman filter solution coincides with the optimal least squares solution which is exactly the posterior mean $\mathbf{m}_{k|k}$. For derivation and further discussion on the matter see, for example, Kalman (1960), Maybeck (1979) and Särkkä (2006).

2.1.2 Rauch–Tung–Striebel Smoother Equations

We take a brief look at fixed-interval optimal smoothing. The purpose of optimal smoothing is to obtain the marginal posterior distribution of the state $\mathbf{x}_k$ at time step k, which is conditional to all the measurements $\mathbf{y}_{1:T}$, where $k \in [1, \ldots, T]$ is a fixed interval.

Similarly as the discrete-time linear Kalman filter gives a closed-form filtering solution, the discrete-time *Rauch–Tung–Striebel Smoother* (RTS) (see, *e.g.*, Rauch *et al.*, 1965; Särkkä, 2006) gives a closed-form solution to the linear smoothing problem. That is, the smoothed state is given as

$$
p(\mathbf{x}_k \mid \mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_{k|T}, \mathbf{P}_{k|T}).
$$

The RTS equations are written so that they utilize the Kalman filtering results $\mathbf{m}_{k|k}$ and $\mathbf{P}_{k|k}$ as a forward sweep, and then perform a backward sweep to update the estimates to match the forthcoming observations (see, *e.g.*, Särkkä, 2006). The forward sweep is already presented in Equations (3) and (4). The smoother's backward sweep may be written as

$$
\begin{aligned}
\mathbf{m}_{k+1|k} &= \mathbf{A}_k \mathbf{m}_{k|k} \\
\mathbf{P}_{k+1|k} &= \mathbf{A}_k \mathbf{P}_{k|k} \mathbf{A}_k^\mathsf{T} + \mathbf{Q}_k \\
\mathbf{C}_k &= \mathbf{P}_{k|k} \mathbf{A}_k^\mathsf{T} \mathbf{P}_{k+1|k}^{-1} \\
\mathbf{m}_{k|T} &= \mathbf{m}_{k|k} + \mathbf{C}_k (\mathbf{m}_{k+1|T} - \mathbf{m}_{k+1|k}) \\
\mathbf{P}_{k|T} &= \mathbf{P}_{k|k} + \mathbf{C}_k (\mathbf{P}_{k+1|T} - \mathbf{P}_{k+1|k}) \mathbf{C}_k^\mathsf{T},
\end{aligned}
\tag{5}
$$

where $\mathbf{m}_{k|T}$ is the smoothed mean and $\mathbf{P}_{k|T}$ the smoother covariance at time step k. The RTS smoother can be seen as a discrete-time forward–backward filter, as the backward sweep utilizes information from the forward filtering sweep. When performing the backward recursion the time steps run from T to 0.

2.2 Non-Linear Kalman Filters

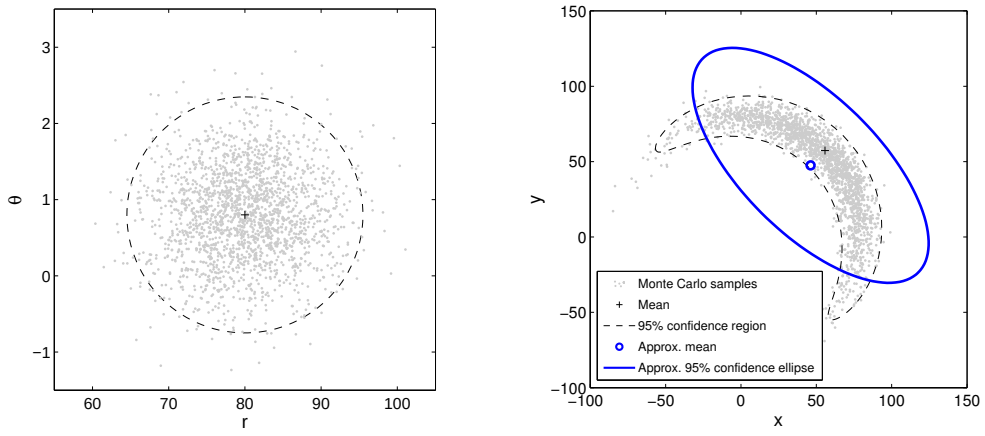
We now move to a more general form of stochastic state-space models, where the dynamic and measurement model functions are not required to be linear, but arbitrary non-linear functions. We consider a non-linear stochastic state-space model of form

$$
\begin{aligned}
\mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1} \\
\mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k,
\end{aligned}
\tag{6}
$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the state, $\mathbf{y}_k \in \mathbb{R}^m$ is the measurement of the state at time step k, $\mathbf{q}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ is the additive Gaussian process noise of the dynamic model, $\mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is the additive Gaussian noise of the measurement model, $\mathbf{f}(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^n$ is the dynamic model function and $\mathbf{h}(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^m$ is the measurement model function.

The linear Kalman filter is heavily founded on the fact that the linear transform preserves the Gaussian nature of the probability distribution throughout the filtering. It is clear that the non-linear transform makes this impossible. There is absolutely no guarantee that the resulting distribution will even faintly resemble anything Gaussian.

Formally this can be seen in the case of a Gaussian random variable $\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{P})$ and an arbitrary non-linear transformation $\mathbf{y} = \mathbf{g}(\mathbf{x})$. If $\mathbf{g}$ is invertible, the probability density of $\mathbf{y}$ is (Gelman, 2004)



(a) Original. (b) Transformed.

Figure 3: This figure illustrates the effect of making a non-linear transform on a bivariate Gaussian distribution and approximating the resulting distribution by a Gaussian.

$$p(\mathbf{y}) = |\mathbf{J}(\mathbf{y})|\,\mathcal{N}(\mathbf{g}^{-1}(\mathbf{y}) \mid \mathbf{m}, \mathbf{P}),$$

where $|\mathbf{J}(\mathbf{y})|$ is the determinant of the Jacobian matrix of the inverse transform $\mathbf{g}^{-1}(\mathbf{y})$. It is not generally possible to handle the resulting distribution $p(\mathbf{y})$ directly. This implies that to find a practicable equivalent to the linear Kalman filter in the case the model functions are non-linear, heavy approximations are needed.

To use the theory of the linear Kalman filter and exploit the convenient features of the Gaussian distribution, the family of non-linear Kalman filters falls back on one basic approximation. All transformed probability distributions are assumed to be approximately Gaussian, that is $p(\mathbf{y}) \approx \mathcal{N}(\mathbf{y} \mid \mathbf{m}', \mathbf{P}')$.

Figure 3 presents an illustrative example of the effects of a non-linear transformation. A bivariate normal distribution $\mathbf{x} = \begin{bmatrix} r & \theta \end{bmatrix}^\mathsf{T} \sim \mathcal{N}(\begin{bmatrix} 80 & 0.8 \end{bmatrix}^\mathsf{T}, \mathrm{diag}(40, 0.4))$ is transformed through the non-linear transformation

$$f(\mathbf{x}) = \begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix},$$

which corresponds to making a change of coordinates from radial to Cartesian coordinates. As we see in Figure 3(b), the true distribution does not resemble a Gaussian. Together with the real mean and confidence region a Gaussian approximation made using 2 000 Monte Carlo samples is shown in the figure. The Gaussian captures the spirit of the true distribution but — obviously — fails to capture the non-linearities of the resulting distribution.

Although the Gaussian approximation based approach is clearly rough, it simplifies the formulation of non-linear filters. The basic idea of non-linear filters is to form Gaussian approximations. How this is done in practice varies a bit. Some well-known non-linear extensions to the classic Kalman filter include: (for more alternatives see, *e.g.*, Särkkä, 2006)

Extended Kalman filter (EKF) relies on the first-order linearization obtained by a Taylor series expansion. EKF was the *de facto* standard of non-linear Kalman filtering for decades (see Jazwinski, 1970).

Statistically linearized Kalman filter (SLF) is a quasi-linear filter where the Gaussian approximation is based on closed-form computations of expected values (see Gelb, 1974).

Unscented Kalman filter (UKF) uses deterministically chosen sigma points to approximate the non-linearities. (see Julier and Uhlmann, 1996; van der Merwe, 2004)

Central difference Kalman filter (CDKF) uses Sterling's polynomial interpolation to approximate the distribution (see Nørgaard *et al.*, 2000).

Gauss–Hermite Kalman filter (GHKF) uses the Gauss–Hermite quadrature rule to solve the Gaussian integrals that arise in the context of filtering (see Ito and Xiong, 2000).

Cubature Kalman filter (CKF) uses a third-degree cubature approximation to solve the Gaussian integrals (see Arasaratnam and Haykin, 2009).

2.3 Assumed Density Estimation

To unify many of the filter variants presented earlier, handling the non-linearities may be brought together to a common formulation. In Gaus-

sian optimal filtering — also called assumed density filtering — the filtering equations follow the assumption that the filtering distributions are indeed Gaussian (Maybeck, 1982; Ito and Xiong, 2000; Särkkä, 2010). The Gaussian approximation is of the form

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) \approx \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_{k|k}, \mathbf{P}_{k|k}),$$

where $\mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_{k|k}, \mathbf{P}_{k|k})$ denotes the multivariate Gaussian distribution with mean $\mathbf{m}_{k|k}$ and covariance $\mathbf{P}_{k|k}$.

The linear Kalman filter equations can now be adapted to the non-linear state-space model of form (6). The prediction step of the non-linear filter can be obtained through calculating the following integrals that approximate the mean $\mathbf{m}_{k|k-1}$ and covariance $\mathbf{P}_{k|k-1}$ given the measurements up to $k-1$:

$$\begin{aligned}
\mathbf{m}_{k|k-1} &= \int \mathbf{f}(\mathbf{x}_{k-1})\, \mathcal{N}(\mathbf{x}_{k-1} \mid \mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})\, \mathrm{d}\mathbf{x}_{k-1} \\
\mathbf{P}_{k|k-1} &= \int \Big(\mathbf{f}(\mathbf{x}_{k-1}) - \mathbf{m}_{k|k-1}\Big) \Big(\mathbf{f}(\mathbf{x}_{k-1}) - \mathbf{m}_{k|k-1}\Big)^{\mathsf{T}} \\
&\qquad \times \mathcal{N}(\mathbf{x}_{k-1} \mid \mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})\, \mathrm{d}\mathbf{x}_{k-1} + \mathbf{Q}_{k-1}
\end{aligned} \tag{7}$$

To form the update step the measurement mean, prediction covariance and cross-covariance between the state and measurement have to be approximated by calculating the integrals

$$\begin{aligned}
\hat{\mathbf{y}} &= \int \mathbf{h}(\mathbf{x}_k)\, \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1})\, \mathrm{d}\mathbf{x}_k \\
\mathbf{S}_k &= \int (\mathbf{h}(\mathbf{x}_k) - \hat{\mathbf{y}})\, (\mathbf{h}(\mathbf{x}_k) - \hat{\mathbf{y}})^{\mathsf{T}}\, \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1})\, \mathrm{d}\mathbf{x}_k + \mathbf{R}_k \\
\mathbf{P}_{xy} &= \int (\mathbf{x}_k - \mathbf{m}_{k|k-1})\, (\mathbf{h}(\mathbf{x}_k) - \hat{\mathbf{y}})^{\mathsf{T}}\, \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1})\, \mathrm{d}\mathbf{x}_k.
\end{aligned} \tag{8}$$

Using the measurement mean $\hat{\mathbf{y}}$ and covariance $\mathbf{S}_k$ together with the cross-covariance between the state and measurement $\mathbf{P}_{xy}$ the *update step* may be written similarly as in the linear Kalman filter:

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{P}_{xy} \mathbf{S}_k^{-1} \\
\mathbf{m}_{k|k} &= \mathbf{m}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}) \\
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^{\mathsf{T}},
\end{aligned}$$

where $\mathbf{K}_k$ is the Kalman gain, $\mathbf{m}_{k|k}$ is the filtered mean at time step k and $\mathbf{P}_{k|k}$ is the respective covariance.

Similarly as the assumed density filtering equations, the Rauch–Tung–Striebel smoother presented in Equation (5) can be generalized to an assumed density form (see Särkkä, 2010; Särkkä and Hartikainen, 2010). The recursion steps for the discrete-time fixed-interval Gaussian assumed density smoother can be written in the form, where we first calculate the Gaussian integrals

$$\begin{aligned}
\mathbf{m}_{k+1|k} &= \int \mathbf{f}(\mathbf{x}_k)\, \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_{k|k}, \mathbf{P}_{k|k})\, \mathrm{d}\mathbf{x}_k \\
\mathbf{P}_{k+1|k} &= \int \Big(\mathbf{f}(\mathbf{x}_k) - \mathbf{m}_{k+1|k}\Big) \Big(\mathbf{f}(\mathbf{x}_k) - \mathbf{m}_{k+1|k}\Big)^{\mathsf{T}}\, \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_{k|k}, \mathbf{P}_{k|k})\, \mathrm{d}\mathbf{x}_k + \mathbf{Q}_k \\
\mathbf{D}_{k,k+1} &= \int \Big(\mathbf{x}_k - \mathbf{m}_{k|k}\Big) \Big(\mathbf{f}(\mathbf{x}_k) - \mathbf{m}_{k+1|k}\Big)^{\mathsf{T}}\, \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_{k|k}, \mathbf{P}_{k|k})\, \mathrm{d}\mathbf{x}_k
\end{aligned} \tag{9}$$

and then the gain term $\mathbf{C}_k$ together with the smoothing result at step k can be calculated as in the linear case (see Equations (5))

$$\begin{aligned}
\mathbf{C}_k &= \mathbf{D}_{k,k+1} \mathbf{P}_{k+1|k}^{-1} \\
\mathbf{m}_{k|T} &= \mathbf{m}_k + \mathbf{C}_k (\mathbf{m}_{k+1|T} - \mathbf{m}_{k+1|k}) \\
\mathbf{P}_{k|T} &= \mathbf{m}_k + \mathbf{C}_k (\mathbf{P}_{k+1|T} - \mathbf{P}_{k+1|k}) \mathbf{C}_k^{\mathsf{T}}.
\end{aligned}$$

The integrals in the filter and smoother equations, (7), (8) and (9), can be solved with basically any suitable analytical or numerical integration method. The contributions on this field have produced a number of filter variations that use different numerical integration methods, for example the Gauss–Hermite Kalman filter (Ito and Xiong, 2000), the Monte Carlo Kalman filter (Kotecha and Djuric, 2003) and the cubature Kalman filter (Arasaratnam and Haykin, 2009), just to mention the most well-known ones.

In the next section we shall take a closer look at numerical integration methods regarding the type of Gaussian integrals that have arisen in the context of assumed Gaussian optimal estimation.

3 Gaussian Cubature Methods

We now consider a multi-dimensional integral of the form

$$\mathrm{I}(\mathbf{f}) = \int_{\mathcal{D}} \mathbf{f}(\mathbf{x})\, w(\mathbf{x})\, \mathrm{d}\mathbf{x}, \tag{10}$$

where $\mathbf{f}(\cdot)$ is a Lebesque integrable arbitrary function, $\mathcal{D} \subseteq \mathbb{R}^n$ is the region of integration, and $w(\mathbf{x})$ is the known non-negative weighting function $w : \mathcal{D} \mapsto \mathbb{R}_+$. In a Gaussian weighted integral, the weight $w(\mathbf{x})$ is a Gaussian density in $\mathcal{D} = \mathbb{R}^n$, and thus it satisfies the non-negativity condition in the entire region.

The integrals that arose in the previous section are of this form. Commonly the closed-form solution to Equation (10) is difficult to obtain, and some numerical approximation method is chosen to calculate the solution. Here we consider a numerical method which tries to find a set of points $\mathbf{x}_i$, $i \in \{1, \ldots, m\}$ and corresponding weights w_i that approximates the integral $\mathrm{I}(\mathbf{f})$ by calculating the value of a weighted sum

$$\mathrm{I}(\mathbf{f}) \approx \sum_{i=1}^{m} w_i \mathbf{f}(\mathbf{x}_i). \tag{11}$$

The methods of finding an appropriate approximation are divided into *product rules* and *non-product rules*. Of these two the former is presented with an emphasis on the Gauss–Hermite cubature rule and the latter with focus on the spherical–radial cubature rule.

3.1 The Gauss–Hermite Quadrature Rule

The Gauss–Hermite quadrature rule (see, *e.g.*, Abramowitz and Stegun, 1964) is a one-dimensional weighted sum approximation method for solving special integrals of form (10) with a Gaussian kernel, with an infinite domain, $\mathcal{D} = \mathbb{R}$. More specifically the Gauss–Hermite quadrature can be applied to integral approximations of form

$$\int_{-\infty}^{\infty} f(x)\, \exp(-x^2)\, \mathrm{d}x \approx \sum_{i=1}^{m} w_i\, f(x_i), \tag{12}$$

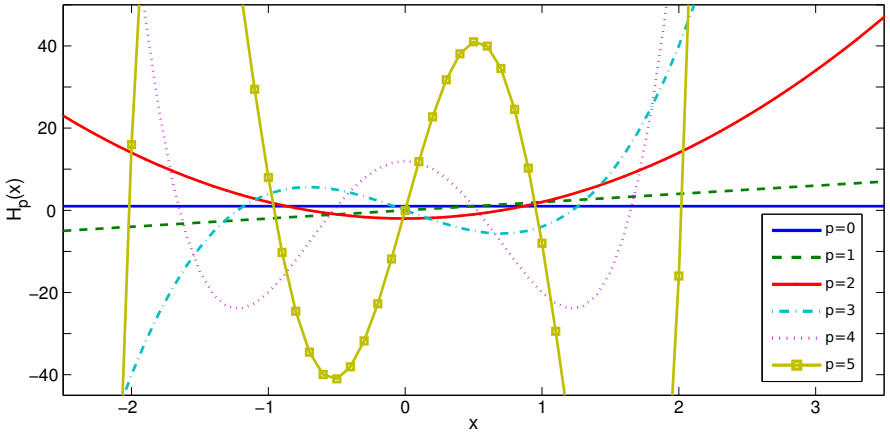


Figure 4: The first six Hermite polynomial curves around origin.

where x_i are the sample points and w_i the associated weights to use for the approximation. The sample points $x_i, i = 1, \ldots, m$, are roots of special orthogonal polynomials called Hermite polynomials. Here we use the so called physicists' Hermite polynomials. The Hermite polynomial of degree p is denoted with $H_p(x)$ (see Abramowitz and Stegun (1964) for details) and can be written as

$$H_p(x) = p! \sum_{m=0}^{\lfloor p/2 \rfloor} \frac{(-1)^m}{m!(p-2m)!} (2x)^{p-2m},$$

where p is the degree and $\lfloor \cdot \rfloor$ denotes the floor operator. The first six Hermite polynomials are shown in Figure 4. The weights w_i are given by

$$w_i = \frac{2^{p-1} p! \sqrt{\pi}}{p^2 [H_{p-1}(x_i)]^2}.$$

The univariate integral approximation needs to be extended to be able to suit the multivariate case. As Wu *et al.* (2006) argue, the most natural approach to grasp a multiple integral is to treat it as a sequence of nested univariate integrals and then use a univariate quadrature rule repeatedly. To extend this one-dimensional integration method to multi-dimensional integrals of form

$$\int_{\mathbb{R}^n} f(\mathbf{x}) \exp(-\mathbf{x}^\mathsf{T}\mathbf{x}) \, \mathrm{d}\mathbf{x} \approx \sum_{i=1}^{m} w_i f(\mathbf{x}_i), \tag{13}$$

we first simply form the one-dimensional quadrature rule with respect to the first dimension, then with respect to the second dimension and so on (Cools, 1997). We get the multidimensional Gauss–Hermite cubature rule by writing

$$\begin{aligned}
&\sum_{i_1} w_{i_1} \int f(x_1^{i_1}, x_2, \ldots, x_n) \exp(-x_2^2 - x_3^2 \ldots - x_n^2) \, \mathrm{d}x_2 \ldots \, \mathrm{d}x_n \\
&= \sum_{i_1, i_2} w_{i_1} w_{i_2} \int f(x_1^{i_1}, x_2^{i_2}, \ldots, x_n) \exp(-x_3^2 \ldots - x_n^2) \, \mathrm{d}x_3 \ldots \, \mathrm{d}x_n \\
&= \sum_{i_1, i_2, \ldots, i_n} w_{i_1} w_{i_2} \cdots w_{i_n} f(x_1^{i_1}, x_2^{i_2}, \ldots, x_n^{i_n}),
\end{aligned}$$

which is basically what we wanted in Equation (13). This gives us the *product rule* that simply extends the one-dimensional quadrature point set of p points in one dimension to a lattice of p^n cubature points in n dimensions. The weights for these Gauss–Hermite cubature points are calculated by the product of the corresponding one-dimensional weights.

Finally, by making a change of variable $\mathbf{x} = \sqrt{2}\sqrt{\boldsymbol{\Sigma}} + \boldsymbol{\mu}$ we get the Gauss–Hermite weighted sum approximation for a multivariate Gaussian integral, where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$ is the covariance of the Gaussian. The square root of the covariance matrix, denoted $\sqrt{\boldsymbol{\Sigma}}$, is a matrix such that $\boldsymbol{\Sigma} = \sqrt{\boldsymbol{\Sigma}}\sqrt{\boldsymbol{\Sigma}}^{\mathsf{T}}$.

$$\int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x}) \, \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \, \mathrm{d}\mathbf{x} \approx \sum_{i_1, i_2, \ldots, i_n} w_{i_1, i_2, \ldots, i_n} f\left(\sqrt{\boldsymbol{\Sigma}} \, \boldsymbol{\xi}_{i_1, i_2, \ldots, i_n} + \boldsymbol{\mu}\right), \quad (14)$$

where the weight $w_{i_1, i_2, \ldots, i_n} = \frac{1}{\pi^{n/2}} w_{i_1} \cdot w_{i_2} \cdots w_{i_n}$ is given by using the normalized one-dimensional weights, and the points are given by the Cartesian product $\boldsymbol{\xi}_{i_1, i_2, \ldots, i_n} = \sqrt{2} \, (x_{i_1}, x_{i_2}, \ldots, x_{i_n})$, where x_i is the ith one-dimensional quadrature point.

The extension of the Gauss–Hermite quadrature rule to an n-dimensional cubature rule by using the product rule lattice approach yields a rather good numerical integration method that is exact for monomials $\prod_{i=1}^{n} x_i^{k_i}$ with $k_i \leq 2p - 1$ (Wu *et al.*, 2006). However, the number of cubature points grows exponentially as the number of dimensions increases. Due to this flaw the rule is not practical in applications with many dimensions. This problem is called the *curse of dimensionality*. The exponentially growing number of evaluation points has been illustrated in Figure 5 where the point lattice for dimensions $n = 1, 2, 3$ is shown.

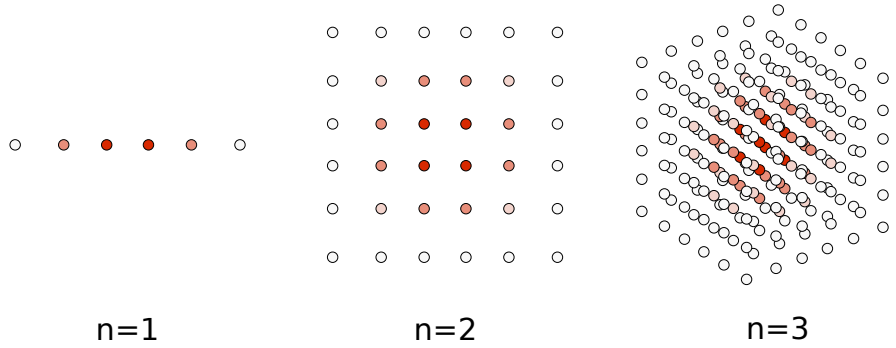


Figure 5: This figure shows the lattice in dimensions $n = 1, 2, 3$ required to perform the $p = 6$ point product rule integral approximation in the Gauss–Hermite cubature rule. The color marks the weight of the cubature point.

3.2 The Spherical–Radial Cubature Rule

The curse of dimensionality causes all product rules to be highly ineffective in integration regions with multiple dimensions. To mitigate this issue, we may seek alternative approaches to solving the integral of form (10). The non-product rules differ from product based solutions by choosing the evaluation points directly from the domain of integration. That is, the points are not simply duplicated from one dimension to multiple dimensions, but directly chosen from the whole domain.

The field of non-product rule based integration methods has spawned many well-known approaches, some of which are: (see Cools, 1997; Arasaratnam, 2009)

Randomized Monte Carlo methods evaluate the integral using a set of equally weighted random points.

Quasi-Monte Carlo methods use deterministically defined methods to provide a hyper-cube region from which the points are randomly drawn.

Lattice rules transform the evaluation point grid to better fit the integration domain and thus diminish the ill-behavior of product rules.

Sparse grids combine a univariate quadrature routine for high-dimensional problems to concentrate most cubature points to areas of interest.

Monomial-based cubature rules are constructed so that they are exact for monomials of a certain degree.

The quest for finding a good integration method comes to setting the balance of choosing a method that (i) yields reasonable accuracy, (ii) requires a small number of function evaluations, and (iii) is easily extendable to high dimensions. The spherical–radial cubature rule to be presented incorporates aspects of all these. The derivation of the third-degree spherical–radial cubature rule mostly follows Arasaratnam (2009) and Wu *et al.* (2006).

3.2.1 Monomial-Based Cubature Approximations

Hereafter we are only interested in integrands of form *non-linear function* $\times$ *Gaussian*. However, the focus is first constrained to an integral of the form

$$\mathrm{I}(\mathbf{f}) = \int_{\mathbb{R}^n} f(\mathbf{x}) \exp(-\mathbf{x}^\mathsf{T}\mathbf{x}) \, \mathrm{d}\mathbf{x}. \tag{15}$$

In Equation (15) the integration domain is defined in Cartesian coordinates. We make a change of variable from $\mathbf{x} \in \mathbb{R}^n$ to spherical coordinates defined by the radius scalar r and the direction vector $\mathbf{y} \in \mathbb{R}^{n-1}$. Let $\mathbf{x} = r\mathbf{y}$ with $\mathbf{y}^\mathsf{T}\mathbf{y} = 1$, so that $\mathbf{x}^\mathsf{T}\mathbf{x} = r^2$ for $r \in [0, \infty)$. The integral can now be written in the spherical–radial coordinate system

$$\mathrm{I}(\mathbf{f}) = \int_0^\infty \int_{S_n} f(r\mathbf{y}) \exp(-r^2) \, r^{n-1} \, \mathrm{d}\sigma(\mathbf{y}) \, \mathrm{d}r. \tag{16}$$

Here S_n is the surface of the sphere defined by $S_n = \{\mathbf{y} \in \mathbb{R}^n \mid \mathbf{y}^\mathsf{T}\mathbf{y} = 1\}$ and $\sigma(\cdot)$ the spherical surface measure of the area element of S_n. We may now write (16) as a *radial integral*

$$\mathrm{I}(\mathbf{f}) = \int_0^\infty \mathrm{S}(r) \, r^{n-1} \exp(-r^2) \, \mathrm{d}r, \tag{17}$$

where $\mathrm{S}(r)$ is defined by the *spherical integral*

$$\mathrm{S}(r) = \int_{S_n} f(r\mathbf{y}) \, \mathrm{d}\sigma(\mathbf{y}). \tag{18}$$

The spherical integral (18) can be seen as a spherical integral with the unit weighting function $w(\mathbf{y}) \equiv 1$. Now the spherical and radial integral may be interpreted separately and computed with the spherical cubature rule and the Gaussian quadrature rule respectively.

3.2.2 Spherical Cubature Rule

A cubature rule is said to be *fully symmetric* if (i) $\mathbf{x} \in \mathcal{D}$ implies $\tilde{\mathbf{x}} \in \mathcal{D}$, where $\tilde{\mathbf{x}}$ is any point obtainable from $\mathbf{x}$ by sign change and permutations of the coordinates of $\mathbf{x}$, and (ii) the points yield the same weight value, $w(\mathbf{x}) = w(\tilde{\mathbf{x}})$ (Arasaratnam, 2009; Cools, 1997).

In other words, in a fully symmetric cubature point set equally weighted points are symmetrically distributed around origin. A point $\mathbf{u}$ is called the *generator* of such a set, if for the components of $\mathbf{u} = \begin{bmatrix} u_1, u_2, \ldots, u_r, 0, \ldots, 0 \end{bmatrix} \in \mathbb{R}^n$, $u_i \geq u_{i+1} > 0$, $i = 1, 2, \ldots, (r-1)$. The point set defined by the generator is denoted $[\mathbf{u}]$. For example, we may denote $[\mathbf{1}] \in \mathbb{R}^2$ to represent the cubature point set

$$\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right\},$$

where the generator is $\begin{bmatrix} 1 & 0 \end{bmatrix}^{\mathsf{T}}$.

We now take a look at Equation (11) in the case where $\mathbf{f}(\cdot)$ is monomial-based. Here the term *monomial* means a product of powers of variables such that the powers are constrained to positive integers. To calculate the integral, a set of cubature points is chosen so that the cubature rule is exact for the degree d or less (for a proof, see Arasaratnam, 2009), as defined by

$$\int_{\mathcal{D}} \mathscr{P}(\mathbf{x})\, w(\mathbf{x})\, \mathrm{d}\mathbf{x} = \sum_{i=1}^{m} w_i \mathscr{P}(\mathbf{x}_i),$$

where $\mathscr{P}$ represents a polynomial, or in this case more specifically a monomial $\mathscr{P}(\mathbf{x}) = x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$ with d_j being non-negative integers and $\sum_{j=1}^{n} d_j \leq d$. In other words, the degree d defines the accuracy of the approximation; the higher the degree, the more accurate the sum.

To find the unknowns of a cubature rule of degree d, a set of moment equations have to be solved. This, however, may not be a simple task with

increasing dimensions and polynomial degree. An m-point cubature rule requires a system of $\frac{(n+d)!}{n!\,d!}$ moment equations to be solved. (Arasaratnam, 2009) For this system to have at least one solution, m must be chosen so that $m \geq \frac{(n+d)!}{(n+1)!\,d!}$. This means that, for example, a cubature rule of degree three in an integration domain of 10 dimensions entails 26 weighted cubature points and yields solving a system of 286 equations.

To reduce the size of the system of equations or the number of needed cubature points Arasaratnam and Haykin (2009) use the *invariant theory* proposed by Sobolev (see Cools, 1997). The invariant theory discusses how to simplify the structure of a cubature rule by using the symmetries of the region of integration. The unit hypercube, the unit hypersphere and the unit simplex all contain some symmetry.

Due to invariance theory (Cools, 1997) the integral (18) can be approximated by a third-degree spherical cubature rule that gives us the sum

$$\int_{S_n} f(r\mathbf{y})\ \mathrm{d}\sigma(\mathbf{y}) \approx w \sum_{i=1}^{2n} \mathbf{f}([\mathbf{u}]_i). \tag{19}$$

The point set $[\mathbf{u}]$ is invariant under permutations and sign changes, which means that a number of $2n$ cubature points are sufficient to approximate the integral. For the above choice, the monomials $y_1^{d_1} y_2^{d_2} \cdots y_n^{d_n}$, with the sum $\sum_{i=1}^{n} d_i$ being an odd integer, are integrated exactly.

To make this rule exact for all monomials up to degree three, we have to require the rule to be exact for the even dimensions $\sum_{i=1}^{n} d_i = \{0, 2\}$. This can be accomplished by solving the unknown parameters for a monomial function of order $n = 0$ and equivalently for a monomial function of order $n = 2$. We consider the two functions $\mathbf{f}(\cdot)$ to be of form $\mathbf{f}(\mathbf{y}) = 1$, and $\mathbf{f}(\mathbf{y}) = y_1^2$. This yields the pair of equations (Arasaratnam, 2009)

$$\begin{aligned} \mathbf{f}(\mathbf{y}) = 1: \qquad & 2nw = \int_{S_n} \mathrm{d}\sigma(\mathbf{y}) = A_n \\ \mathbf{f}(\mathbf{y}) = y_1^2: \qquad & 2wu^2 = \int_{S_n} y_1^2\ \mathrm{d}\sigma(\mathbf{y}) = \frac{1}{n} A_n, \end{aligned}$$

where A_n is the surface area of the n-dimensional unit sphere. Solving these equations yields $u^2 = 1$ and $w = \frac{A_n}{2n}$. Therefore the cubature points can be chosen so that they are located at the intersection of the unit sphere and its axes.

3.2.3 Radial Cubature Rule

We now concentrate on the radial integral defined in Equation (17). This integral can be transformed to a familiar Gauss–Laguerre form (see Abramowitz and Stegun, 1964) by making another change of variable, $t = r^2$, which yields

$$\int_0^\infty \mathrm{S}(r)\, r^{n-1} \exp(-r^2)\, \mathrm{d}r = \frac{1}{2} \int_0^\infty \tilde{\mathrm{S}}(t)\, t^{\frac{n}{2}-1} \exp(-t)\, \mathrm{d}t, \tag{20}$$

where $\tilde{\mathrm{S}}(t) = \mathrm{S}(\sqrt{t})$. This integral may be approximated by the Gauss–Laguerre quadrature by

$$\frac{1}{2} \int_0^\infty \tilde{\mathrm{S}}(t)\, t^{\frac{n}{2}-1} \exp(-t)\, \mathrm{d}t = \sum_{i=1}^{m} w_i\, \tilde{\mathrm{S}}(t_i), \tag{21}$$

where t_i is the ith root of *Laguerre polynomial* $L_m(t)$ and the weights w_i are given by (Abramowitz and Stegun, 1964)

$$w_i = \frac{t_i}{(m+1)^2 (L_{m+1}(t_i))^2}.$$

A first-degree Gauss–Laguerre rule is exact for $\tilde{\mathrm{S}}(t) = \{1, t\}$ (or equivalently $\mathrm{S}(r) = \{1, r^2\}$). It is not exact for $\mathrm{S}(r) = \{r, r^3\}$, but due to the properties of the spherical cubature rule presented in the previous section, the combined spherical–radial rule vanishes by symmetry for all odd degree polynomials. Hence, to have the spherical–radial rule to be exact for all polynomials up to degree three in $\mathbf{x} \in \mathbb{R}^n$ it is sufficient to use the first degree Gauss–Laguerre rule of the form (Arasaratnam, 2009)

$$\int_0^\infty \tilde{\mathrm{S}}_i(t)\, t^{\frac{n}{2}-1} \exp(-t)\, \mathrm{d}t = w_1\, \tilde{\mathrm{S}}_i(t_1), \quad i = \{0, 1\},$$

where $\tilde{\mathrm{S}}_0(t) = 1$ and $\tilde{\mathrm{S}}_1(t) = t$. Now by solving the moment equations we get

$$\begin{aligned} \tilde{\mathrm{S}}_0(t) = 1: \qquad & w_1 = \int_0^\infty t^{\frac{n}{2}-1} \exp(-t)\, \mathrm{d}t = \Gamma(\frac{n}{2}) \\ \tilde{\mathrm{S}}_1(t) = t: \qquad & w_1 t_1 = \int_0^\infty t^{\frac{n}{2}} \exp(-t)\, \mathrm{d}t = \frac{n}{2}\Gamma(\frac{n}{2}). \end{aligned}$$

The first-degree Gauss–Laguerre approximation is constructed using the point $t_1 = \frac{n}{2}$ and the weight $w_1 = \Gamma(\frac{n}{2})$, where $\Gamma(\cdot)$ is the Gamma function. The final radial form approximation can be written using Equation (20) in the form

$$\int_0^\infty \mathrm{S}(r)\, r^{n-1} \exp(-r^2)\, \mathrm{d}r \approx \frac{1}{2}\,\Gamma\left(\frac{n}{2}\right) \mathrm{S}\left(\sqrt{\frac{n}{2}}\right). \tag{22}$$

3.2.4 Combining the Spherical and Radial Rules

We now again consider the integral

$$\mathrm{I}(\mathbf{f}) = \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x}) \exp(-\mathbf{x}^\mathsf{T}\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{23}$$

Now we have an approximation for the spherical integral in Equation (19), where the third-degree rule is acquired by the cubature point set $[\mathbf{1}]$ and weight $\frac{A_n}{2n}$. Here the surface area A_n of the $n-1$ hypersphere equals $2\frac{\pi^{n/2}}{\Gamma(n/2)}$, where $\Gamma(\cdot)$ is the Gamma function.

By applying the results derived for the spherical and radial integral, we may combine Equations (19) and (22) to construct a third-degree cubature approximation for (23), which yields the elegant solution

$$\mathrm{I}(\mathbf{f}) \approx \frac{\sqrt{\pi^n}}{2n} \sum_{i=1}^{2n} \mathbf{f}\left(\sqrt{\frac{n}{2}}[\mathbf{1}]_i\right).$$

Next we may consider the problem of numerically computing a standard Gaussian integral of form

$$\mathrm{I}_N(\mathbf{f}) = \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\, \mathcal{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{I})\, \mathrm{d}\mathbf{x}, \tag{24}$$

where $\mathcal{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{I})$ is the zero-mean Gaussian normal distribution with unit covariance. Per definition of the Gaussian density function,

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right),$$

we can get a third-degree approximation for Equation (24) by adding the scaling factor $(2\pi)^{-n/2}$ to the previously acquired approximation. Thus we get the approximation for (24)

$$\mathrm{I}_N(\mathbf{f}) \approx \frac{1}{2n} \sum_{i=1}^{2n} \mathbf{f}\left(\sqrt{n}[\mathbf{1}]_i\right).$$

Finally, by making a change of variable $\mathbf{x} = \sqrt{\boldsymbol{\Sigma}}\mathbf{y} + \boldsymbol{\mu}$ we get the approximation of an arbitrary integral that is of form *non-linear function* $\times$ *Gaussian*. It can be written as

$$\int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\, \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})\, \mathrm{d}\mathbf{x} \approx \sum_{i=1}^{2n} w_i \mathbf{f}\left(\sqrt{\boldsymbol{\Sigma}}\, \boldsymbol{\xi}_i + \boldsymbol{\mu}\right),$$

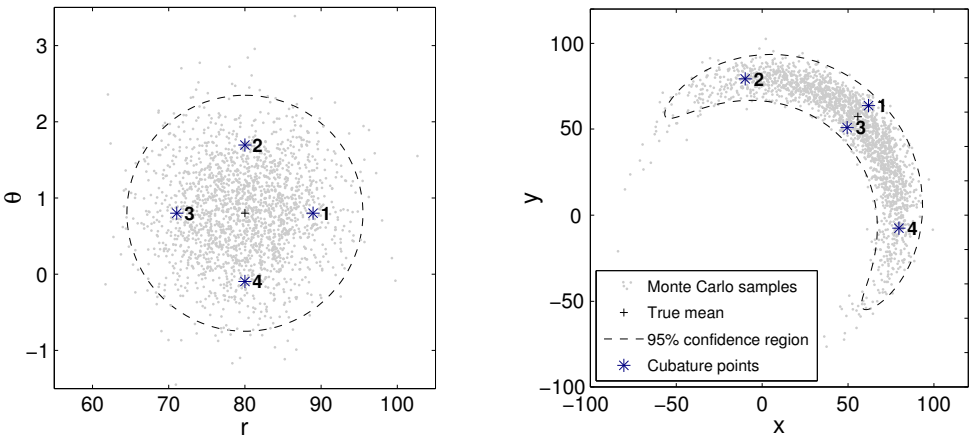
where the cubature points are $\boldsymbol{\xi}_i = \sqrt{n}[\mathbf{1}]_i$, the corresponding (equal) weights $w_i = \frac{1}{2n}$ and the points $[\mathbf{1}]_i$ from the intersections between the Cartesian axes and the n-dimensional unit hypersphere.

4 Cubature Optimal Estimation

Previously we have defined a way of forming Gaussian approximation based Kalman filters through non-linear transformations. Additionally, two different cubature integration rules have been presented for calculating the integrals needed for solving the approximate Gaussian.

4.1 Transformations

In practice both the Gauss–Hermite and the spherical–radial cubature rule fall into the category of *sigma point* based transforms — and the corresponding filters into the category of sigma point filters (van der Merwe, 2004). The sigma point approach means that the resulting approximation is formed with the help of a few well-chosen points that are propagated through the non-linear transformation and then used to match the Gaussian to.



(a) Original. (b) Transformed.

Figure 6: An illustrative example of the sigma point way of forming approximations. Here the cubature rule forms an approximation based on four sigma points.

In Figure 6 an illustrative example of the spherical–radial cubature transformation is presented. The non-linear transformation is the same as presented earlier in Section 2.2. The sigma points, or actually the cubature

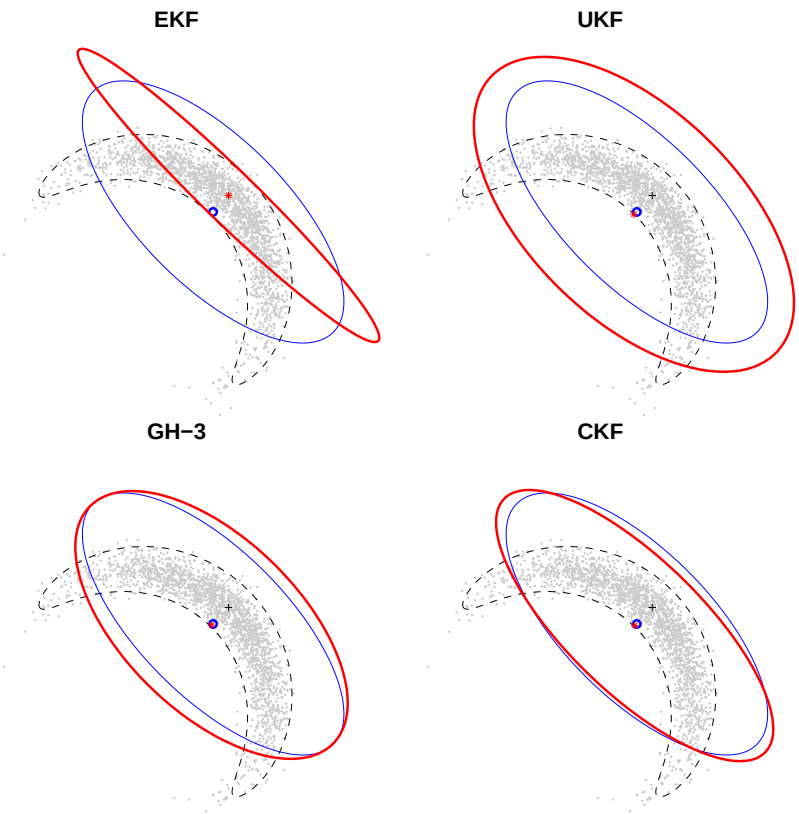


Figure 7: This figure illustrates the differences between the extended transformation, unscented transformation ($\alpha = 0.5$, $\beta = 2$, $\kappa = 1$), the 3rd degree ($p = 3$) Gauss–Hermite transformation and the cubature transformation. The thin solid blue circle in each figure is the exact normal approximation.

points, are drawn from the intersection points of the unit circle and the axes, and they have then been transformed to match the mean and covariance of the original Gaussian as presented in Section 3.2.

The four points are propagated through the transformation. The result can be seen in Figure 6(b). The arithmetic mean or point of mass of the cubature points return an approximation of the Gaussian mean. This was defined in Equation (7) in Section 2.3. Similarly the covariance can be calculated by evaluating the second integral.

Of the earlier presented filtering algorithms the unscented Kalman filter (UKF) is perhaps the most well known sigma point method. Therefore we compare the Gauss–Hermite (GH) and cubature methods (CKF) against the unscented Kalman filter. For comparison reasons, also the local linearization approach of the extended Kalman filter (EKF) is provided. For

brevity the transformations are entitled with the abbreviation of the corresponding filter.

Figure 7 shows four approximations of the exact Gaussian fit. The local approach of the EKF clearly differs from the three others. Here the unscented transform is done using parameter values $\alpha = 0.5$, $\beta = 2$ and $\kappa = 3 - n$, where $n = 2$ (see Wan and van der Merwe, 2001). GH-3 refers to the third-degree Gauss–Hermite rule.

4.2 Spherical–Radial Optimal Estimation

4.2.1 Cubature Kalman Filter

We now once more turn our interest back to the non-linear optimal estimation problem. Previously, in Section 2.3, we went through the *assumed density* approach to estimating the filtering and smoothing solution. In the context of assumed density estimation several integrals need to be solved to be able to form the needed approximations for the non-linear transformations.

Two cubature based numerical integration methods were presented in Section 3. Earlier we also briefly reviewed the transformations accomplished by these methods.

Now, to form the cubature Kalman filter (CKF) algorithm we need to combine the assumed density equations with the spherical–radial cubature rule. The algorithm is presented in Listing 1. The CKF algorithm follows the description by Arasaratnam and Haykin (2009).

The recursive iteration starts at step $k = 0$ with a prior distribution $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{m}_{0|0}, \mathbf{P}_{0|0})$. The cubature points are drawn with the generator $[\mathbf{1}]$ and then propagated with the help of prior mean and covariance. The matrix square root is the lower triangular Cholesky factor, so that $\mathrm{chol}(\mathbf{A}) = \mathbf{L}$, where $\mathbf{L}\mathbf{L}^\mathsf{T} = \mathbf{A}$. The predictive distribution is calculated by solving two integrals from Equations (7).

The update step is calculated similarly as the prediction step. The integrals that are evaluated are presented in Equations (8). The measurement at step k is denoted $\mathbf{y}_k$.

Listing 1: The cubature Kalman filter (CKF) algorithm. At time $k = 1, \ldots, T$ assume the posterior density function $p(\mathbf{x}_{k-1} \mid \mathbf{y}_{k-1}) = \mathcal{N}(\mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})$ is known.

Prediction step:

1. Draw cubature points $\boldsymbol{\xi}_i, i = 1, \ldots, 2n$ from the intersections of the n-dimensional unit sphere and the Cartesian axes. Scale them by $\sqrt{n}$. That is
$$\boldsymbol{\xi}_i = \begin{cases} \sqrt{n}\, e_i & , i = 1, \ldots, n \\ -\sqrt{n}\, e_{i-n} & , i = n+1, \ldots, 2n \end{cases}$$

2. Propagate the cubature points. The matrix square root is the lower triangular cholesky factor.
$$\mathbf{X}_{i,k-1|k-1} = \sqrt{\mathbf{P}_{k-1|k-1}}\boldsymbol{\xi}_i + \mathbf{m}_{k-1|k-1}$$

3. Evaluate the cubature points with the dynamic model function
$$\mathbf{X}^*_{i,k|k-1} = \mathbf{f}(\mathbf{X}_{i,k-1|k-1}).$$

4. Estimate the predicted state mean
$$\mathbf{m}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \mathbf{X}^*_{i,k|k-1}.$$

5. Estimate the predicted error covariance
$$\mathbf{P}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \mathbf{X}^*_{i,k|k-1}\mathbf{X}^{*\mathsf{T}}_{i,k|k-1} - \mathbf{m}_{k|k-1}\mathbf{m}^{\mathsf{T}}_{k|k-1} + \mathbf{Q}_{k-1}.$$

Update step:

1. Draw cubature points $\boldsymbol{\xi}_i, i = 1, \ldots, 2n$ from the intersections of the n-dimensional unit sphere and the Cartesian axes. Scale them by $\sqrt{n}$.

2. Propagate the cubature points.
$$\mathbf{X}_{i,k|k-1} = \sqrt{\mathbf{P}_{k|k-1}}\boldsymbol{\xi}_i + \mathbf{m}_{k|k-1}$$

3. Evaluate the cubature points with the help of the measurement model function
$$\mathbf{Y}_{i,k|k-1} = \mathbf{h}(\mathbf{X}_{i,k|k-1}).$$

4. Estimate the predicted measurement
$$\hat{\mathbf{y}}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \mathbf{Y}_{i,k|k-1}.$$

5. Estimate the innovation covariance matrix
$$\mathbf{S}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \mathbf{Y}_{i,k|k-1}\mathbf{Y}^{\mathsf{T}}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1}\hat{\mathbf{y}}^{\mathsf{T}}_{k|k-1} + \mathbf{R}_k.$$

6. Estimate the cross-covariance matrix
$$\mathbf{P}_{xy,k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} \mathbf{X}_{i,k-1|k-1}\mathbf{Y}^{\mathsf{T}}_{i,k|k-1} - \mathbf{m}_{k|k-1}\hat{\mathbf{y}}^{\mathsf{T}}_{k|k-1}.$$

7. Estimate the Kalman gain
$$\mathbf{K}_k = \mathbf{P}_{xy,k|k-1}\mathbf{S}^{-1}_{k|k-1}.$$

8. Estimate the updated state
$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}).$$

9. Estimate the error covariance
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_{k|k-1}\mathbf{K}^{\mathsf{T}}_k.$$

4.2.2 Cubature Rauch–Tung–Striebel Smoother

The fixed-interval cubature Rauch–Tung–Striebel smoother uses the assumed density equations similarly as the cubature Kalman filter. The integrals that need to be evaluated were presented in Equations (9). Listing 2 presents the spherical–radial based cubature Rauch–Tung–Striebel smoother algorithm.

The backward iteration of the algorithm starts at step $k = T$ where the filtering and smoothing results coincide. Hereafter the recursion progresses through all time steps using both the filtering solution and the information from the already smoothed steps.

Listing 2: The cubature Rauch–Tung–Striebel smoother (CRTS) algorithm. Assume the filtering result mean $\mathbf{m}_{k|k}$ and covariance $\mathbf{P}_{k|k}$ are known together with the smoothing result $p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{m}_{k+1|T}, \mathbf{P}_{k+1|T})$.

1. Draw cubature points $\boldsymbol{\xi}_i, i = 1, \ldots, 2n$ from the intersections of the n-dimensional unit sphere and the Cartesian axes. Scale them by $\sqrt{n}$. That is
$$\boldsymbol{\xi}_i = \begin{cases} \sqrt{n}\, e_i & , i = 1, \ldots, n \\ -\sqrt{n}\, e_{i-n} & , i = n+1, \ldots, 2n \end{cases}$$

2. Propagate the cubature points
$$\mathbf{X}_{i,k|k} = \sqrt{\mathbf{P}_{k|k}}\boldsymbol{\xi}_i + \mathbf{m}_{k|k}.$$

3. Evaluate the cubature points with the dynamic model function
$$\mathbf{X}^*_{i,k+1|k} = \mathbf{f}(\mathbf{X}_{i,k|k}).$$

4. Estimate the predicted state mean
$$\mathbf{m}_{k+1|k} = \frac{1}{2n}\sum_{i=1}^{2n} \mathbf{X}^*_{i,k+1|k}.$$

5. Estimate the predicted error covariance
$$\mathbf{P}_{k+1|k} = \frac{1}{2n}\sum_{i=1}^{2n} \mathbf{X}^*_{i,k+1|k}\mathbf{X}^{*\mathsf{T}}_{i,k+1|k} - \mathbf{m}_{k+1|k}\mathbf{m}^{\mathsf{T}}_{k+1|k} + \mathbf{Q}_k.$$

6. Estimate the cross-covariance matrix
$$\mathbf{D}_{k,k+1} = \frac{1}{2n}\sum_{i=1}^{2n} (\mathbf{X}_{i,k|k} - \mathbf{m}_{k|k}) \left(\mathbf{X}^*_{i,k+1|k} - \mathbf{m}_{k+1|k}\right)^{\mathsf{T}}.$$

7. Estimate the gain term
$$\mathbf{C}_k = \mathbf{D}_{k,k+1}\mathbf{P}^{-1}_{k+1|k}.$$

8. Estimate the smoothed state mean
$$\mathbf{m}_{k|T} = \mathbf{m}_{k|k} + \mathbf{C}_k(\mathbf{m}_{k+1|T} - \mathbf{m}_{k+1|k}).$$

9. Estimate the smoothed state covariance
$$\mathbf{P}_{k|T} = \mathbf{P}_{k|k} + \mathbf{C}_k(\mathbf{P}_{k+1|T} - \mathbf{P}_{k+1|k})\mathbf{C}^{\mathsf{T}}_k.$$

Listing 3: The Gauss–Hermite Kalman filter (GHKF) algorithm of degree p. At time $k = 1, \ldots, T$ assume the posterior density function $p(\mathbf{x}_{k-1} \mid \mathbf{y}_{k-1}) = \mathcal{N}(\mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})$ is known.

Prediction step:

1. Find the roots $x_i, i = 1, \ldots, p$, of the Hermite polynomial $H_p(x)$.
2. Calculate the corresponding weights
$$w_i = \frac{2^{p-1} p!}{p^2 [H_{p-1}(x_i)]^2}.$$
3. Use the product rule to expand the points to a n-dimensional lattice of p^n points $\boldsymbol{\xi}_i, i = 1, \ldots, p^n$, with corresponding weights.
4. Propagate the cubature points. The matrix square root is the lower triangular cholesky factor.
$$\mathbf{X}_{i,k-1|k-1} = \sqrt{2\mathbf{P}_{k-1|k-1}}\boldsymbol{\xi}_i + \mathbf{m}_{k-1|k-1}$$
5. Evaluate the cubature points with the dynamic model function
$$\mathbf{X}^*_{i,k|k-1} = \mathbf{f}(\mathbf{X}_{i,k-1|k-1}).$$
6. Estimate the predicted state mean
$$\mathbf{m}_{k|k-1} = \sum_{i=1}^{p^n} w_i \mathbf{X}^*_{i,k|k-1}.$$
7. Estimate the predicted error covariance
$$\mathbf{P}_{k|k-1} = \sum_{i=1}^{p^n} w_i \mathbf{X}^*_{i,k|k-1} \mathbf{X}^{*\mathsf{T}}_{i,k|k-1} - \mathbf{m}_{k|k-1}\mathbf{m}^{\mathsf{T}}_{k|k-1} + \mathbf{Q}_{k-1}.$$

Update step:

1. Repeat steps 1–3 from earlier to get the p^n cubature points and their weights.
2. Propagate the cubature points.
$$\mathbf{X}_{i,k|k-1} = \sqrt{2\mathbf{P}_{k|k-1}}\boldsymbol{\xi}_i + \mathbf{m}_{k|k-1}$$
3. Evaluate the cubature points with the help of the measurement model function
$$\mathbf{Y}_{i,k|k-1} = \mathbf{h}(\mathbf{X}_{i,k|k-1}).$$
4. Estimate the predicted measurement
$$\hat{\mathbf{y}}_{k|k-1} = \sum_{i=1}^{p^n} w_i \mathbf{Y}_{i,k|k-1}.$$
5. Estimate the innovation covariance matrix
$$\mathbf{S}_{k|k-1} = \sum_{i=1}^{p^n} w_i \mathbf{Y}_{i,k|k-1} \mathbf{Y}^{\mathsf{T}}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1}\hat{\mathbf{y}}^{\mathsf{T}}_{k|k-1} + \mathbf{R}_k.$$
6. Estimate the cross-covariance matrix
$$\mathbf{P}_{xy,k|k-1} = \sum_{i=1}^{p^n} w_i \mathbf{X}_{i,k-1|k-1} \mathbf{Y}^{\mathsf{T}}_{i,k|k-1} - \mathbf{m}_{k|k-1}\hat{\mathbf{y}}^{\mathsf{T}}_{k|k-1}.$$
7. Estimate the Kalman gain
$$\mathbf{K}_k = \mathbf{P}_{xy,k|k-1}\mathbf{S}^{-1}_{k|k-1}.$$
8. Estimate the updated state
$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}).$$
9. Estimate the error covariance
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_{k|k-1}\mathbf{K}^{\mathsf{T}}_k.$$

4.3 Gauss–Hermite Optimal Estimation

Listing 3 shows the Gauss–Hermite Kalman filter algorithm that uses Hermite polynomials to solve the Gaussian integrals. The degree of approximation can be controlled by choosing the number of the Gauss–Hermite quadrature points p.

The algorithm is very similar to the cubature Kalman filter algorithm presented in Listing 1. The only disparities come from the differences between the Gauss–Hermite and spherical–radial integral evaluation. The Gauss–Hermite rule was presented in Section 3.

The Gauss–Hermite Rauch–Tung–Striebel smoother algorithm is presented in Listing 4. The assumed density form Gaussian integrals from Equations (9) are evaluated with the Gauss–Hermite rule.

Listing 4: The Gauss–Hermite Rauch–Tung–Striebel smoother (GHRTS) algorithm of degree p. Assume the filtering result mean $\mathbf{m}_{k|k}$ and covariance $\mathbf{P}_{k|k}$ are known together with the smoothing result $p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{m}_{k+1|T}, \mathbf{P}_{k+1|T})$.

1. Repeat steps 1–3 in Listing 3 to get the p^n cubature points and their weights.
2. Propagate the cubature points
$$\mathbf{X}_{i,k|k} = \sqrt{2\mathbf{P}_{k|k}}\boldsymbol{\xi}_i + \mathbf{m}_{k|k}.$$
3. Evaluate the cubature points with the dynamic model function
$$\mathbf{X}^*_{i,k+1|k} = \mathbf{f}(\mathbf{X}_{i,k|k}).$$
4. Estimate the predicted state mean
$$\mathbf{m}_{k+1|k} = \sum_{i=1}^{p^n} w_i \mathbf{X}^*_{i,k+1|k}.$$
5. Estimate the predicted error covariance
$$\mathbf{P}_{k+1|k} = \sum_{i=1}^{p^n} w_i \mathbf{X}^*_{i,k+1|k} \mathbf{X}^{*\mathsf{T}}_{i,k+1|k} - \mathbf{m}_{k+1|k}\mathbf{m}^{\mathsf{T}}_{k+1|k} + \mathbf{Q}_k.$$
6. Estimate the cross-covariance matrix
$$\mathbf{D}_{k,k+1} = \frac{1}{2n} \sum_{i=1}^{2n} (\mathbf{X}_{i,k|k} - \mathbf{m}_{k|k}) \left(\mathbf{X}^*_{i,k+1|k} - \mathbf{m}_{k+1|k}\right)^{\mathsf{T}}.$$
7. Estimate the gain term
$$\mathbf{C}_k = \mathbf{D}_{k,k+1}\mathbf{P}^{-1}_{k+1|k}.$$
8. Estimate the smoothed state mean
$$\mathbf{m}_{k|T} = \mathbf{m}_{k|k} + \mathbf{C}_k(\mathbf{m}_{k+1|T} - \mathbf{m}_{k+1|k}).$$
9. Estimate the smoothed state covariance
$$\mathbf{P}_{k|T} = \mathbf{P}_{k|k} + \mathbf{C}_k(\mathbf{P}_{k+1|T} - \mathbf{P}_{k+1|k})\mathbf{C}^{\mathsf{T}}_k.$$

5 Case Studies

5.1 Target Tracking of a Maneuvering Target

Target tracking applications are often used to demonstrate the effect of Kalman filtering. This comes natural as the states of the dynamic system are easy to associate with physical properties like position and velocity. In this example we consider a coordinated turn model where we track a maneuvering target on a two-dimensional plane with the help of two sensors.

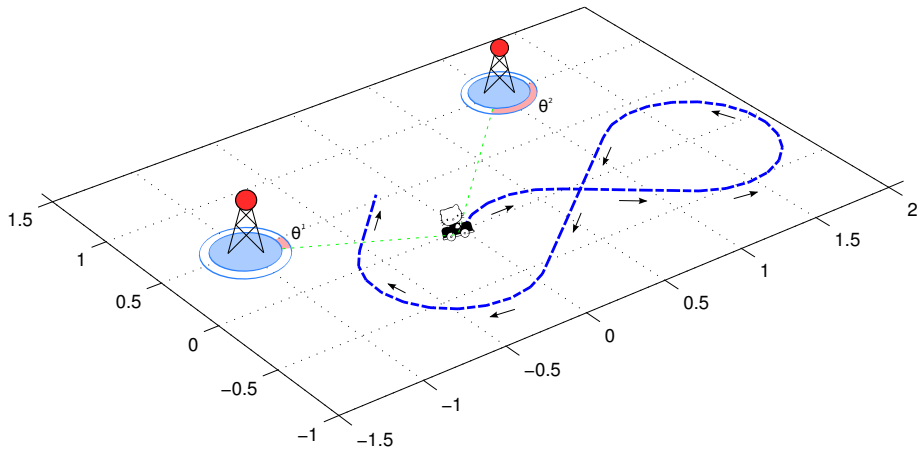


Figure 8: A sketch of the experiment setting used by the coordinated turn model with bearings only tracking. The actual trajectory of the vehicle is shown in dashed blue. The towers illustrate the sensor positions.

Figure 8 shows the basic test setup. We have a vehicle and its trajectory on a plane. Two sensors track the position of the vehicle by returning noisy measurements of angular direction of the target. As each sensor return only an angle θ_i the tracking model is called a *bearings only tracking* model (Bar-Shalom *et al.*, 2001).

The dynamic model is a *coordinated turn* model. These types of models are often used in air traffic control as civilian aircraft have two basic modes of flight: (i) Straight flight with constant speed and course, (ii) Maneuvering, when the course is changed by turning. (Bar-Shalom *et al.*, 2001) For brevity and easing illustrative representation we consider a case where

the vehicle is constrained to move on a two-dimensional plane — as shown in Figure 8.

The dynamic model of the coordinated turn model (Bar-Shalom *et al.*, 2001) is

$$\mathbf{x}_k = \begin{bmatrix} 1 & 0 & \frac{\sin(\omega\Delta t)}{\omega} & -\left(\frac{1-\cos(\omega\Delta t)}{\omega}\right) & 0 \\ 0 & 1 & \frac{1-\cos(\omega\Delta t)}{\omega} & \frac{\sin(\omega\Delta t)}{\omega} & 0 \\ 0 & 0 & \cos(\omega\Delta t) & -\sin(\omega\Delta t) & 0 \\ 0 & 0 & \sin(\omega\Delta t) & \cos(\omega\Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{q}_{k-1}, \tag{25}$$

where the state of the target is $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dot{x}_1 & \dot{x}_2 & \omega \end{bmatrix}^\mathsf{T}$. The coordinates are (x_1, x_2), the velocities $(\dot{x}_1, \dot{x}_2)$ and the turn rate is ω. The additive noise of the dynamic model is $\mathbf{q}_{k-1} \sim \mathcal{N}(0, \mathbf{Q}_{k-1})$. The coordinated turn model is necessarily non-linear if the turn rate is not a known constant. A similar model is used by Särkkä and Hartikainen (2010) and Arasaratnam and Haykin (2009).

The observations of the state are obtained through two sensors that measure the angles θ^i between the target and the sensor. The non-linear measurement model for each sensor i can be written (Särkkä and Hartikainen, 2010) as

$$\theta_k^i = \arctan\left(\frac{x_{2,k} - s_y^i}{x_{1,k} - s_x^i}\right) + r_k^i, \tag{26}$$

where (s_x^i, s_y^i) is the position of the sensor i, and $r_k^i \sim \mathcal{N}(0, \sigma_\theta^2)$ is the measurement noise.

5.1.1 Experiment Settings

Both the dynamic and the measurement models are non-linear. We use four different non-linear filters and smoothers to track the movement of the vehicle: the extended Kalman filter (EKF), the unscented Kalman filter (UKF), the third-degree Gauss–Hermite Kalman filter (GHKF) and the spherical–radial rule based cubature Kalman filter (CKF). We also compare smoothing results for the four RTS smoothers corresponding to

the filters. The unscented Kalman filter parameters are $\alpha = 0.5$, $\beta = 2$ and $\kappa = 3 - n$, where $n = 5$ (see Wan and van der Merwe, 2001).

The simulated route is discretized into 500 time steps with $\Delta t = 0.01$. The dynamic model noise covariance is

$$\mathbf{Q}_{k-1} = \begin{bmatrix} q_c\Delta t^3/3 & 0 & q_c\Delta t^2/2 & 0 & 0 \\ 0 & q_c\Delta t^3/3 & 0 & q_c\Delta t^2/2 & 0 \\ q_c\Delta t^2/2 & 0 & q_c\Delta t & 0 & 0 \\ 0 & q_c\Delta t^2/2 & 0 & q_c\Delta t & 0 \\ 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}, \tag{27}$$

where $q_c = 0.1$. The measurement noise of the angular measurement was assumed to be $\sigma_\theta = 0.05$. The two sensors tracking the movement are positioned at $(-1, 0.5)$ and $(1, 1)$.

5.1.2 Results

The filters and smoothers were compared by running them over 100 independent simulations. The comparison was done by calculating the root mean square error (RMSE) for the position, velocity and angular components separately and averaging them over all the runs. In addition, the number of function evaluations used by each method gives a rough estimate of the computational effectiveness of each algorithm.

Table 1: This table shows the RMSE values calculated by averaging the results of 100 independent runs. The error is separately shown for position, velocity and turn rate components. Additionally, the number of required function evaluations is shown on far right.

	Position		Velocity		Turn rate		Function
	Filter	Smoother	Filter	Smoother	Filter	Smoother	evaluations
EKF	0.0696	0.0237	0.5099	0.1262	42.0297	22.1552	6
UKF	0.0669	0.0233	0.4939	0.1234	41.8408	22.1308	11
GHKF	0.0669	0.0233	0.4939	0.1234	41.8330	22.1266	243
CKF	0.0669	0.0233	0.4943	0.1235	41.8406	22.1263	10

Figure 9 shows a realization of the true trajectory (in dashed black) together with the CKF filter and smoother estimates (in solid blue and red) of the trajectory. The uncertainty of the estimate is illustrated with the help of the 95 % confidence regions of the estimates. The results of other filters are not shown as the results are practically identical.

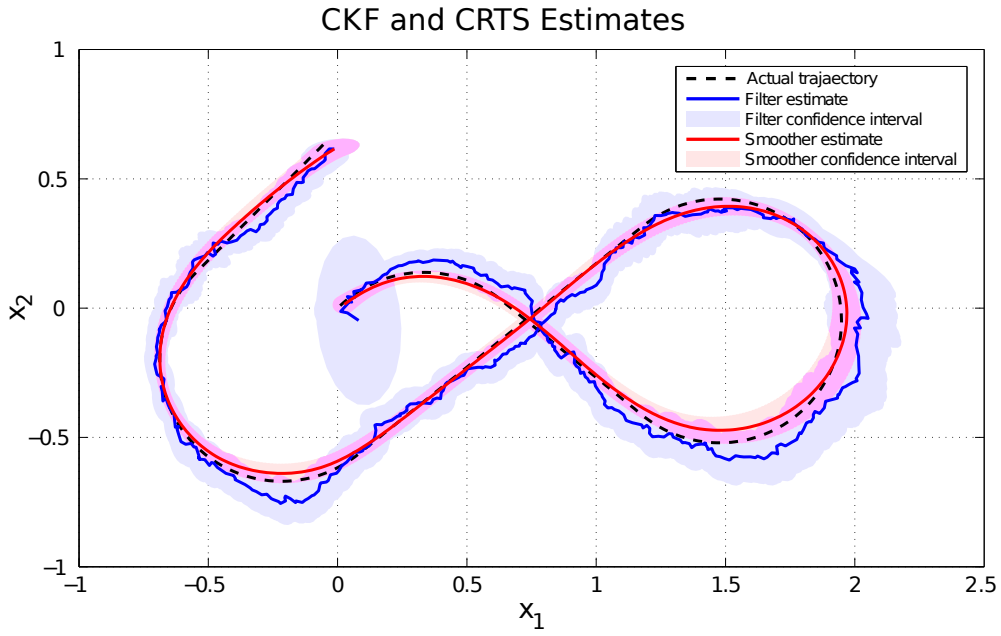


Figure 9: A realization of the tracking problem. The colored patches represent the 95 % confidence intervals.

Table 1 contains the averaged RMSE values. As can be seen, the local linearization used by EKF causes it to be slightly less accurate than the other three filters and smoothers. Notably the results of UKF, GHKF and CKF are practically equally good.

The main difference between these three methods is the execution time. Table 1 shows the number of function evaluations per filter. It is clearly visible that the Gauss–Hermite Kalman filter requires tens of times more execution time than the other three filters. This is due to the curse of dimensionality. GHKF requires $3^5 = 243$ function evaluations on each integral calculation, which is a lot compared to UKF ($2 \cdot 5 + 1 = 11$ evaluations) and CKF ($2 \cdot 5 = 10$ evaluations). EKF requires only one evaluation together with the evaluation of its n-dimensional Jacobian matrix ($1 + n = 6$ evaluations).

5.2 Training a MLP Neural Network

Singhal and Wu (1989) present an interesting application for multilayer perceptron (MLP) neural networks. They use a global extended Kalman filter (GEKF) approach to train a neural network so that it identifies shapes of four different colors in a rectangular region. The four-region classification problem has since often (see, *e.g.*, Wan and van der Merwe, 2001; van der Merwe, 2004) been used as an example of Kalman filter based training and to help comparing different training methods.

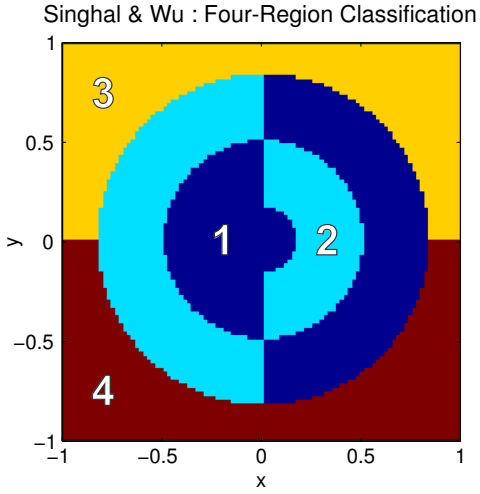


Figure 10: The shape consisting of four regions that Singhal and Wu used. In this figure the square has been discretized to a 100×100 lattice.

The four-region classification problem relates to a figure where a square-formed area is divided in two and a couple of interlocked circular regions are attached to the figure. These geometrical shapes have been illustrated in Figure 10. The shapes can be seen as a function that maps x and y co-ordinates to a color between 1–4 by returning a binary vector.

The classification problem is to train a 2-10-10-4 feedforward MLP neural network with the help of the Figure 10 to return a figure with the same shapes. The network has two input nodes, the co-ordinates $(x, y) \in [-1, 1] \times [-1, 1]$. In addition, the input layer has one bias term. The network uses two hidden layers with 10 nodes each. The network has four output nodes. For in-depth explanations of a MLP neural network structure see, for example, Haykin (1999).

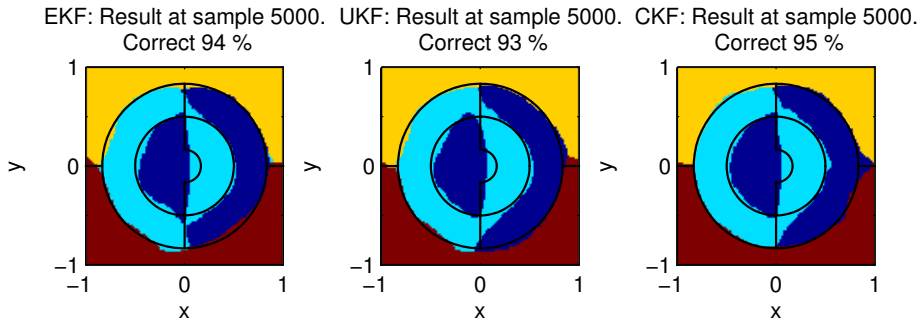


Figure 11: This figure shows an example of network output after training the network with 5000 random samples. The true figure is illustrated by the outlines.

5.2.1 Experiment Settings

The network contains 170 weights altogether, which defines the dimension of the dynamic model. The estimation problem may be written with similar notation as the previously discussed filtering problems. The state space representation is

$$\begin{aligned} \mathbf{w}_{k+1} &= \mathbf{w}_k + \mathbf{q}_k \\ \mathbf{d}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{w}_k) + \mathbf{r}_k, \end{aligned}$$

where the weights $\mathbf{w}_k \in \mathbb{R}^{170}$ correspond to a stationary process. The desired output $\mathbf{d}_k \in \mathbb{R}^4$ corresponds to a non-linear observation on $\mathbf{w}_k$. The dynamic model function is linear but the measurement model function is non-linear. We use non-linear tanh activation as the activation function of the MLP neural network.

The square is discretized into a 100×100 uniform grid. From this grid a number of $n_s = 5000$ random samples are drawn for training. The initial state is drawn from a uniform distribution, $\mathbf{w}_0 \sim \mathcal{U}(-2, 2)$. The initial covariance is set to a diagonal matrix with $\sigma = 0.01$ along the diagonal. The diagonal noise term covariances are set to the values $\mathbf{Q}_0 = 10^{-5}\,\mathbf{I}$ and $\mathbf{R}_0 = 10^{-3}\,\mathbf{I}$. They are exponentially annealed toward the values $\mathbf{Q}_{n_s} = 10^{-9}\,\mathbf{I}$ and $\mathbf{R}_{n_s} = 10^{-7}\,\mathbf{I}$ respectively as the training progresses (Wan and van der Merwe, 2001).

We use three different non-linear Kalman filters for parallel training: EKF, UKF and CKF. All filters use the same initial states and samples during the training steps. For the unscented Kalman filter we use parameter

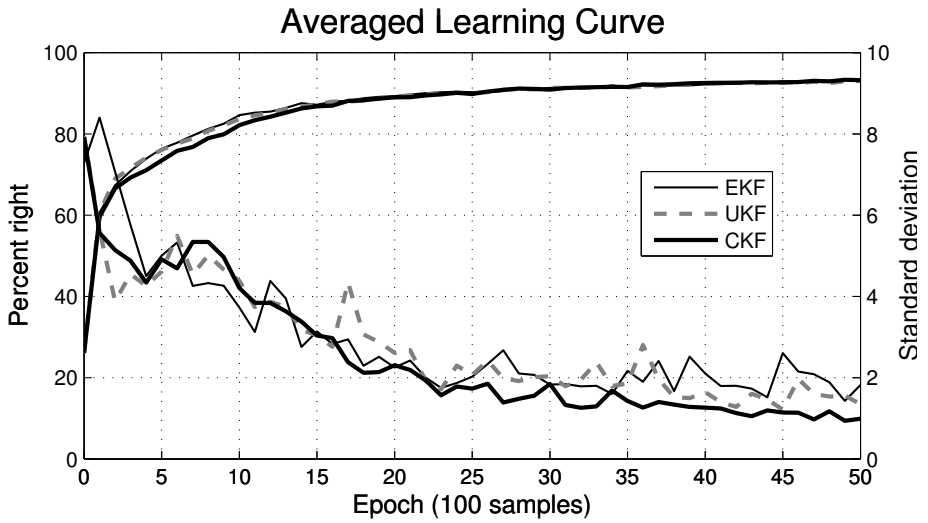


Figure 12: The learning curve averaged over 50 independent runs shows small variation between the methods. Each run consisted of 5000 samples to teach the network (1 epoch = 100 random samples).

values $\alpha = 0.5$, $\beta = 2$ and $\kappa = 3 - n$, where $n = 170$ (see Wan and van der Merwe, 2001). Notably this is not an optimized method for training MLP neural networks, but a challenging estimation problem for non-linear Kalman filters.

5.2.2 Results

By running the three parallel Kalman filters over 5000 samples the network weights converge on average to a solution that is able to return a representation of the four region problem that is accurate up to about 93 %. An example of the network outputs after the training is presented in Figure 11.

The random samples and the initial state cause variation in the performance of the three filters. The parallel training of the three neural networks was run 50 times and the average convergence was studied. Figure 12 shows the average convergence properties of the three different filtering methods. The left vertical axis shows the accuracy of the returned network output after a certain number of samples. The right axis shows the standard deviation of the result.

The EKF and UKF solutions resemble each other, whereas the CKF solution converges a bit slower but results in a good solution with small variance.

6 Discussion

The two experiment cases that were presented in the previous section bring forth the most notable properties of the cubature optimal estimation methods studied in this thesis.

As the results in the target tracking example suggest, the differences between the UKF, GHKF and CKF methods are small or non-existent in practical examples. The local linearization approach used by EKF is clearly good enough for this particular estimation problem, but it does not preserve the Gaussian nature of the approximation — as seen in Section 4.1.

The Gauss–Hermite Kalman filter (GHKF) and the similarly formulated smoother suffer from the curse of dimensionality. This was shown in Section 3.1. Consequently it requires a magnitude of function evaluations that makes it impossible to implement in cases with many dimensions. The tracking example shows that the GHKF is clearly slow and the excessive number of calculations do not yield any improvements in the results.

The curse of dimensionality is even clearer in the four region classification problem, as the third-degree Gauss–Hermite rule would require $3^{170} \approx 10^{81}$ function evaluations making it impossible to implement. Yet, the Gauss–Hermite method makes it easy to control the degree of approximation. The number of one-dimensional quadrature points defines the degree.

The spherical–radial rule provides the cubature Kalman filter (CKF) with the accuracy of the GHKF but less computational complexity. With $2n$ function evaluations it is a clear rival of the unscented Kalman filter ($2n + 1$ evaluations). In fact, it can be shown that the CKF falls back to a special case of the UKF with parameters $\alpha = 1$, $\beta = 0$ and $\kappa = 0$. This is shown in Appendix A.

Additionally, Wan and van der Merwe (2001) show a relation between the GHKF and the UKF. For the scalar case the unscented transform with $\alpha = 1$, $\beta = 0$ and $\kappa = 2$ coincides with the three-point Gauss–Hermite quadrature rule.

The cubature Kalman filter shares many of the good properties of the unscented Kalman filter. They both are derivative-free as no closed-form derivatives are required or continuity requirements are set.

The spherical–radial rule that was derived in this thesis is a third-degree rule. Arasaratnam and Haykin (2009) discuss the need for higher-degree cubature rules. A higher-degree rule yields more accuracy only if the integrand is well-behaved in the sense of approximations of higher-degree polynomials, and the weighting function follow the Gaussian density exactly. Arasaratnam and Haykin (2009) state that the use of higher-degree cubature rules in the design of the CKF may often sabotage its performance.

7 Conclusions

In this thesis the non-linear optimal estimation framework was presented with the help of an assumed density approach. The Gaussian integrals that arise in this setting were solved using two different cubature integration methods.

Both of these methods use deterministically chosen sigma points to form the desired approximation. The Gauss–Hermite rule used a simple product rule method to fill the multidimensional space with cubature points, whereas the spherical–radial rule uses invariant theory to diminish the number of points by utilizing symmetries.

The most important remarks regarding the Gauss–Hermite rule based filter (GHKF) and smoother (GHRTS) and the spherical–radial rule based cubature Kalman filter (CKF) and smoother (CRTS) are:

- Both the Gauss–Hermite and the spherical–radial cubature rule are derivative free. No closed-form representations or continuity requirements are needed. This is desirable in problems with considerable non-linearities.
- The Gauss–Hermite cubature rule suffers from the curse of dimensionality as it entails p^n cubature points, where n is the state-vector dimension. Even though, in problems with only a few state-space dimensions the ease of controlling the degree of the Gaussian approximation makes the Gauss–Hermite method appealing.
- The spherical–radial rule uses $2n$ cubature points. As this is the theoretical lower bound of points for a third-degree rule, the spherical–radial rule based cubature Kalman filter may be considered as an optimal approximation to the non-linear Bayesian filter under the Gaussian assumption.
- The spherical–radial cubature rule can be seen as a special case of the unscented transform with parameters $\alpha = 1$, $\beta = 0$ and $\kappa = 0$. Yet the well-justified numerical properties of the cubature Kalman filter make it a new and welcomed refinement to the unscented transform.

All in all the cubature integration methods provide a different perspective to existing methods and justify the use of certain parameters in the UKF setting.

References

Abramowitz, M. and Stegun, I. A. (1964). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, ninth Dover printing, tenth GPO edition.

Arasaratnam, I. (2009). *Cubature Kalman Filtering: Theory & Applications*. PhD thesis, ECE Department, McMaster University.

Arasaratnam, I. and Haykin, S. (2009). Cubature Kalman filters. *IEEE Transactions on Automatic Control*, 54(6):1254–1269.

Bar-Shalom, Y., Li, X., and Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience.

Cools, R. (1997). Constructing cubature formulae: The science behind the art. *Acta Numerica*, 6:1–54.

Deisenroth, M., Huber, M., and Hanebeck, U. (2009). Analytic moment-based Gaussian process filtering. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 225–232. ACM.

Gelb, A. (1974). *Applied Optimal Estimation*. The MIT press.

Gelman, A. (2004). *Bayesian Data Analysis*. CRC press, second edition.

Grewal, M. and Andrews, A. (2001). *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley-Intersciece, second edition.

Haykin, S. (1999). *Neural Networks: a Comprehensive Foundation*. Prentice Hall PTR Upper Saddle River, NJ, USA.

Ito, K. and Xiong, K. (2000). Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45(5):910–927.

Jazwinski, A. (1970). *Stochastic Processes and Filtering Theory*. Academic Press.

Julier, S. and Uhlmann, J. (1996). A general method for approximating nonlinear transformations of probability distributions. *Dept. of Engineering Science, University of Oxford, Tech. Rep*.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Kotecha, J. and Djuric, P. (2003). Gaussian particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2592–2601.

Maybeck, P. S. (1979). *Stochastic Models, Estimation and Control*, volume 1. Academic Press.

Maybeck, P. S. (1982). *Stochastic Models, Estimation and Control*, volume 2. Academic Press.

Nørgaard, M., Poulsen, N., and Ravn, O. (2000). New developments in state estimation for nonlinear systems. *Automatica*, 36(11):1627–1638.

Rauch, H., Tung, F., and Striebel, C. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450.

Singhal, S. and Wu, L. (1989). Training multilayer perceptrons with the extended Kalman algorithm. In *Advances in Neural Information Processing Systems 1*, pages 133–140. Morgan Kaufmann Publishers Inc.

Särkkä, S. (2006). *Recursive Bayesian inference on stochastic differential equations*. Doctoral dissertion, Helsinki University of Technology.

Särkkä, S. (2010). Bayesian estimation of time-varying systems – Discrete-time systems. Written material for the course S-114.4202, Aalto University, School of Science and Technology.

Särkkä, S. and Hartikainen, J. (2010). On Gaussian optimal smoothing of non-linear state space models. *Accepted for publication in IEEE Transactions on Automatic Control*, 55(8):1938–1941.

van der Merwe, R. (2004). *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, Oregon Health & Science University.

Wan, E. and van der Merwe, R. (2001). The unscented Kalman filter. In Haykin, S., editor, *Kalman Filtering and Neural Networks*. Wiley-Interscience.

Welch, G. and Bishop, G. (1995). An introduction to the Kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC.

Wu, Y., Hu, D., Wu, M., and Hu, X. (2006). A numerical-integration perspective on Gaussian filters. *IEEE Transactions on Signal Processing*, 54(8):2910–2921.

Appendix A:
The Spherical–Radial Rule as a Special Case of the Unscented Transform

The unscented transform uses a fixed number of deterministically chosen sigma points to capture the mean and covariance of the distribution to be approximated. The *unscented transform* to form the Gaussian approximation is done the following way (Wan and van der Merwe, 2001; Särkkä, 2006):

1. Form the matrix of sigma points

$$\mathbf{X} = \begin{bmatrix} \mathbf{m} & \dots & \mathbf{m} \end{bmatrix} + \sqrt{n+\lambda} \begin{bmatrix} \mathbf{0} & \sqrt{\mathbf{P}} & -\sqrt{\mathbf{P}} \end{bmatrix},$$

   where λ is a scaling parameter which is defined in terms of the parameters α and κ as

$$\lambda = \alpha^2(n+\kappa) - n.$$

2. Propagate the sigma points through the non-linear function $\mathbf{f}(\cdot)$

$$\mathbf{Y}_i = \mathbf{f}(\mathbf{X}_i), \qquad i = 1, \dots, 2n+1,$$

   where $\mathbf{X}_i$ and $\mathbf{Y}_i$ denote the ith column of matrices $\mathbf{X}$ and $\mathbf{Y}$ respectively.

3. Estimates of the mean $\hat{\mathbf{m}}$ and covariance $\hat{\mathbf{P}}$ of the transformed variable can be acquired through evaluating the following sums

$$\begin{aligned} \hat{\mathbf{m}} &= \sum_{i=1}^{2n+1} W_{i-1}^{(m)} \mathbf{Y}_i \\ \hat{\mathbf{P}} &= \sum_{i=1}^{2n+1} W_{i-1}^{(c)} (\mathbf{Y}_i - \hat{\mathbf{m}})(\mathbf{Y}_i - \hat{\mathbf{m}})^{\mathsf{T}}, \end{aligned}$$

   where the constant weights $W_i^{(m)}$ and $W_i^{(c)}$ are defined as

$$
\begin{aligned}
W_0^{(m)} &= \lambda/(n+\lambda) \\
W_0^{(c)} &= \lambda/(n+\lambda) + (1-\alpha^2+\beta) \\
W_i^{(m)} &= 1/\{2(n+\lambda)\}, \qquad i = 1,\ldots,2n \\
W_i^{(c)} &= 1/\{2(n+\lambda)\}, \qquad i = 1,\ldots,2n.
\end{aligned}
$$

If we require the scaling parameter λ to be equal to zero, we have to choose the other parameters to be $\alpha = \pm 1$ and $\kappa = 0$. Now the sigma points used by the unscented transform are equivalent to the spherical–radial rule cubature points except that instead of $2n$ points there are $2n+1$ points. The additional point is located exactly at the mean $\mathbf{m}$.

If we set the parameter $\beta = 0$, the additional mean point is zero-weighted and can be discarged. Now the remaining $2n$ points are equally weighted with the weights $W_i^{(m)} = W_i^{(c)} = 1/2n$, which is exactly the case in the spherical–radial rule based cubature transformation.

This means that the cubature transform coincide with the unscented transform when the unscented transform is done with parameters $\alpha = \pm 1$, $\beta = 0$ and $\kappa = 0$.

Appendix B: Suomenkielinen yhteenveto

Kubatuuri-integrointimenetelmien käyttö epälineaarisessa Kalman-suodatuksessa ja silotuksessa

Optimaalinen estimointi tarkoittaa dynaamisen järjestelmän todellisen tilan hakemista käyttäen kohinaisia ja mahdollisesti epäsuorasti saatuja mittaustuloksia. Sovelluskohteita löytyy runsaasti eri aloilta. Optimaalisen estimoinnin suodatus- ja silotusmenetelmiä hyödynnetään muun muassa navigoinnissa (GPS-paikannus), signaalinkäsittelyssä, taloudessa ja koneoppimisessa (ks. esim. Särkkä, 2006).

Tässä työssä tarkastellaan diskreettiaikaisia dynaamisia systeemejä, jotka voidaan esittää tilayhtälömallina

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1} \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k, \end{aligned} \tag{1}$$

jossa $\mathbf{x}_k \in \mathbb{R}^n$ on tila ja $\mathbf{y}_k \in \mathbb{R}^m$ on mittaus aika-askeleella $k = 1, \ldots, T$. Mallin dynamiikka tulee kuvauksesta $\mathbf{f}(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^n$ ja havainnot mittausmallista $\mathbf{h}(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^m$. Mallin normaalijakautuneeksi oletettu kohina on puhtaasti additiivista. Prosessikohina on muotoa $\mathbf{q}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ ja mittauskohina $\mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. (Kalman, 1960; Bar-Shalom ym., 2001)

Termi *suodatus* tarkoittaa aika-askeleella k tilan $\mathbf{x}_k$ marginaalijakauman selvittämistä annettuna kaikki aikahetkeen mennessä saadut mittaukset $\mathbf{y}_{1:k}$. Tätä jakaumaa voidaan merkitä $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$. *Silotus* puolestaan viittaa tilanteeseen, jossa käytettävissä on myös aika-askeletta k seuraavia mittauksia. Marginaalijakaumaa voidaan tällöin merkitä $p(\mathbf{x}_k \mid \mathbf{y}_{1:T})$, jossa $T > k$.

Tilanteissa, joissa yhtälöt (1) ovat lineaarisia, suodatusongelma voidaan ratkaista suljetussa muodossa käyttäen Kalman-suodinta (Kalman, 1960), ja vastaavalle silotusongelmalle saadaan ratkaisu käyttäen Rauch–Tung–Striebel (RTS) -silotusta (Rauch ym., 1965).

Epälineaarinen tiladynamiikka- ja mittausmalli hankaloittavat tilannetta, sillä kuvaukset eivät tällöin säilytä jakaumien gaussista luonnetta. Epälineaarisia suotimia on vuosien varrella esitelty useita. Näistä tunnetuin on ensimmäisen asteen derivaattojen antamaan linearisaatioon

perustuva *laajennettu Kalman-suodin* (Extended Kalman filter, EKF) (ks. Jazwinski, 1970). Muita tunnettuja lähestymistapoja tarjoavat esimerkiksi *tilastollisesti linearisoitu Kalman-suodin* (Statistically linearized Kalman filter, SLF) (ks. Gelb, 1974) sekä deterministisesti valittuihin sigma-pisteisiin nojaava *hajustamaton Kalman-suodin* (Unscented Kalman filter, UKF) (ks. Julier ja Uhlmann, 1996; van der Merwe, 2004).

Monien epälineaaristen suotimien esittäminen matemaattisesti voidaan yhtenäistää kirjoittamalla ne oletetun tiheyden muotoon (Assumed density filtering, ADF), jossa suodatus- ja silotusyhtälöt kirjoitetaan olettaen, että estimoitava jakauma todella noudattaisi gaussista normaalijakaumaa (Maybeck, 1982; Ito ja Xiong, 2000; Särkkä, 2010). Gaussinen approksimaatio on tällöin muotoa

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) \approx \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_{k|k}, \mathbf{P}_{k|k}),$$

jossa $\mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_{k|k}, \mathbf{P}_{k|k})$ on moniulotteinen normaalijakauma keskiarvolla $\mathbf{m}_{k|k}$ ja kovarianssilla $\mathbf{P}_{k|k}$. Käyttämällä tätä muotoa suodatus- ja silotusyhtälöiden ratkaiseminen kiteytyy muutaman integraalin laskemiseen. Näissä integraaleissa integroitava funktio on käytännössä epälineaarisen funktion ja normaalijakauman tulo eli niin sanottu gaussinen integraali.

Suodatus- ja silotusyhtälöissä esiintyvät gaussiset integraalit voidaan ratkaista millä tahansa analyyttisellä tai numeerisella integrointimenetelmällä. Yksi monista lähestymistavoista integraalien ratkaisemiseen on kubatuurisääntöjen käyttö. Moniulotteinen kubatuurisääntö yleistää yksiulotteiset kvadratuurisäännöt useaan ulottuvuuteen. Kubatuurisäännöt pyrkivät approksimoimaan integraalia painotetulla summalla

$$\int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})\, w(\mathbf{x})\, \mathrm{d}\mathbf{x} \approx \sum_{i=1}^{m} w_i\, \mathbf{f}(\mathbf{x}_i),$$

jossa $\mathbf{f}(\cdot)$ on epälineaarinen kuvaus, $w(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}_+$ painofunktio ja w_i diskreetit painot. Yksiulotteinen Gauss–Hermite-kvadratuurisääntö soveltuu gaussisten integraalien ratkaisemiseen yhdessä ulottuvuudessa. Tämä kvadratuurisääntö voidaan laajentaa n-ulotteiseksi kubatuurisäännöksi ottamalla n-kertainen karteesinen tulo yksiulotteisesta kvadratuuripistejoukosta. (Wu ym., 2006) Tätä kutsutaan tulosäännöksi, ja sen huonona puolena on kubatuuripisteiden määrän eksponentiaalinen kasvu ulottuvuuksien määrän kasvaessa, mikä tunnetaan myös dimensionaalisuuden kirouksena. Gauss–Hermite-kubatuurisäännön

käyttäminen vaatii siis p^n kubatuuripistettä, missä p on yksiulotteisen Gauss–Hermite-kvadratuurisäännön pisteiden määrä ja n ulottuvuuksien määrä.

Tapa välttää tarvittavien pisteiden määrän kasvua on hyödyntää integrointialueen symmetriaa. Käyttämällä Sobolevin invarianssiteoriaa voidaan kubatuuripisteiden määrää vähentää (ks. Cools, 1997). Arasaratnamin ja Haykinin (2009) esittelemä pallo- ja sädesymmetrinen kubatuurisääntö perustuu integraalin jakamiseen kahtia; sisäkkäiset pallonkuoret ja säteittäinen integraali ratkaistaan erikseen. Yhdistämällä nämä säännöt saadaan kolmannen asteen monomien mielessä optimaalinen integrointisääntö, jossa $2n$ samoin painotettua kubatuuripistettä on valittu yksikköpallon ja karteesisten akselien risteymäkohdista ja skaalattu ulottuvuuksien määrän neliöjuurella.

ADF-yhtälöiden esitystapaa käyttäen voidaan Gauss–Hermite-säännön ja pallo–säde-symmetrisen säännön avulla muodostaa Gauss–Hermite-Kalman-suodin (GHKF) sekä (pallo–säde-symmetrinen) kubatuuri-Kalman-suodin (CKF). Vastaavasti voidaan muotoilla näitä kahta integrointisääntöä hyödyntävät silotusmenetelmät.

Kubatuuri-Kalman-suodin jakaa hajustamattoman Kalman-suotimen kanssa joukon ominaisuuksia. Kummatkin nojaavat deterministisiin sigma-pisteisiin, vaativat ulottuvuuksien määrään suoraan verrannollisesti evaluointipisteitä (UKF: $2n + 1$, CKF: $2n$), eivätkä ne kaipaa derivaattoja suljetussa muodossa. Itse asiassa liitteessä A osoitetaan, että CKF palautuu UKF-suotimeksi, jossa on käytetty parametreja $\alpha = 1$, $\beta = 0$ ja $\kappa = 0$.

Työssä tehdyissä simulaatioissa näkyy, että käytännössä eri suotimien erot jäävät hyvin pieniksi. Esitellyistä menetelmistä voidaan todeta, että laskennallisesti raskaan GHKF:n etuna on sen tarjoama helppo tapa kontrolloida approksimaation astetta. CKF puolestaan perustelee tiettyjen parametrien valintaa UKF:ssä ja tarjoaa uuden näkökulman sigma-pisteiden valitsemiseen.