Multi-Modal Inference in Time Series

Non-Linear State Space Models

Master-Thesis von Sanket Kamthe aus Pune,India

Januar 2014

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Informatik/ETiT

Signal Processing Group

Multi-Modal Inference in Time Series
Non-Linear State Space Models

Vorgelegte Master-Thesis von Sanket Kamthe aus Pune,India

1. Gutachten: Dr.-Ing. Marc P Deisenroth
2. Gutachten: Prof. Dr. Jan Peters
3. Gutachten: Prof. Dr.-Ing. Henning Puder

Tag der Einreichung:

Bitte zitieren Sie dieses Dokument als:
URN: urn:nbn:de:tuda-tuprints-12345
URL: http://tuprints.ulb.tu-darmstadt.de/1234

Dieses Dokument wird bereitgestellt von tuprints,
E-Publishing-Service der TU Darmstadt
http://tuprints.ulb.tu-darmstadt.de
tuprints@ulb.tu-darmstadt.de

Erklärung zur Master-Thesis

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den January 31, 2014

(Sanket Kamthe)

Abstract

Multi-modal densities appear frequently in time series and practical applications. However, they cannot be represented by common state estimators, such as the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF), which additionally suffer from the fact that uncertainty is often not captured sufficiently well, which can result in incoherent and divergent tracking performance. In this thesis, we address these issues by devising a non-linear filtering algorithm where densities are represented by Gaussian mixture models, whose parameters are estimated in closed form. The filtered results can be further improved by a backward pass or smoothing. However, the optimal backward filter does not offer a closed form solution and, hence, approximations are needed. We propose a novel algorithm for smoothing in non-linear dynamical systems with multi-modal beliefs. The resulting method exhibits a superior performance on typical benchmarks.

Acknowledgements

First of all, I would thank my supervisors Dr.-Ing. Marc P Deisenroth and Prof. Jan Peters for accepting me as an external student, and Prof. Henning Puder for his support as an internal supervisor.

I have learned a lot from my interactions with Marc, and I am grateful for his continued support even after moving to London. I admire him for his hands-on approach to supervision, especially for encouraging some of my outlandish ideas. I thank him for his suggestions on the approximate inference algorithms and his patience to help me draft my first research paper. Both, the paper and his suggestions constitute a substantial part of this thesis.

I thank Prof. Jan Peters for his support during my thesis with not only technical supervision and financial assistance but also guidance on host of other questions I keep directing towards him.

I acknowledge the partial financial support for my thesis by Akademisches Auslandsamt at Technische Universität Darmstadt

I thank Felix Schmitt, Dominik Notz, Marco Ewerton and Roberto Calandra for making my stay in the lab enjoyable and for indulging in discussions from abstract mathematics to utter nonsense. My floor-mates Aishwarya Sharma and Divya Yenna helped me to cope up with absolute lows in my personal life and their support allowed me to concentrate on my thesis, and, hence, I would like to thank them.

This space would be too small to express my gratitude to my parents, as I cannot thank them enough for their continued support throughout my studies without which I could not have reached this stage in my career.

Contents

1 Introduction

The Hawk-Eye (Owens et al., 2003) technology has revolutionised the way we watch tennis. How do they generate ball trajectory ? A path traced by a dynamical system (ball) is called trajectory of that dynamical system. We express trajectory as a times series, that is our trace is over time, as we are interested in predicting the trajectory (over time). The dynamical systems are often described by differential equation in time. In the Figure 1 Hawk-Eye technology generates the trajectory of the ball. The proprietary system was originally developed for cricket in 2001 and now is included in English Premier League as well.

The Hawk-Eye uses a set of cameras that capture the motion of the ball and based on the model of dynamical system we estimate the position of ball in a tennis court (Owens et al., 2003). Tracking a moving object based on observations is a classic example of estimation or filtering (Bar-Shalom et al., 2004). Filtering is a signal processing term for estimation of a true signal from a noisy data (we filter noise). The filtering algorithms to estimate the true signal based on noisy time series data have wide applications from biomedicine to rocket science.

In fact, the first (linear) filter for tracking non-stationary systems was proposed by Kalman (Kalman, 1960) for National Aeronautics and Space Administration (NASA). The Kalman filter, the Hawk-Eye system and this work share a common theme, they estimate the latent state of a system (ball) from sequential noisy observations. Filtering and estimation in a non-linear dynamical system has been studied extensively and hence, has resulted in unique terminology. We use the tennis ball tracking as an example to establish terminology and as a visual aid [1].

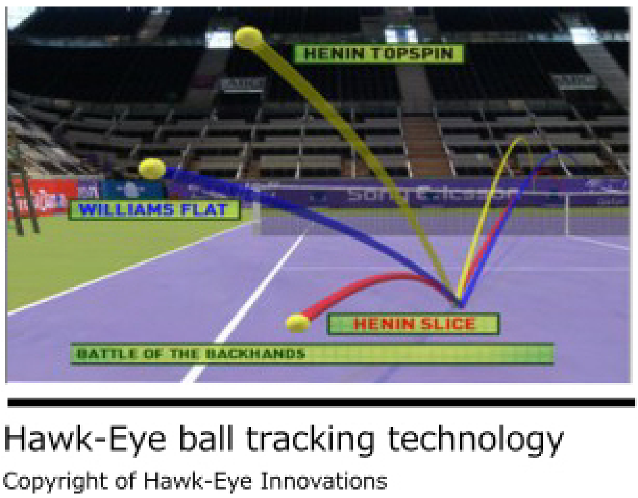


Figure 1.1.: Ball tracking: The shaded trace represents estimated trajectory of the ball. The trajectory can be represented as a time series describing time evolution of a dynamical system.(For definitions of the terms see text)

We define the ball and the forces acting on it (wind, gravity etc.) as a *'system'*. We wish to track the ball, i.e. the position of the ball in space and its velocity, which tell us the current *'state'* of our system, as our state changes over time we define state by *'state variables'*. Now we cannot actually measure these state variables directly without disturbing the ongoing match, so we *'observe'* the *'latent'* state of ball. For technical reasons, it is easier to measure distance and the angle the ball makes with respect to observer, we call these measurements as *'observed variables'*. Now, we aim to predict how the ball will move in future based on the measurements of the ball by the observer. We may be able to approximately model the

[1] The ball tracking is used only as a visual aid to interpret abstract terminology. The focus of thesis is not tracking a tennis ball, however, the algorithms proposed here can be used for tracking.

tennis court, e.g. hard court or clay court to predict the velocity of ball once it hits the surface. However, the challenge of accurately modelling the impact forces and the approximations in model introduce *'uncertainty'* in our system and we can only *'estimate'* the position of the ball.

How we account for the 'uncertainty' introduced by the approximate model of our complex real life system is central theme of this thesis. The unknown noise makes the task even more challenging, we model noise a Gaussian with known variance. In this report, we restrict ourselves to the Gaussian noise model and non-linear dynamical systems. See Section 1.1.1 for details on scope of the work.

An optimal algorithm for a linear dynamical system with Gaussian noise was proposed by Kalman (Kalman, 1960). In such systems, the Gaussianity allows us to derive the recursive filtering equations in closed-form. In contrast, for a non-linear system Gaussian uncertainties may become non-Gaussian due to the non-linear transform. Hence, we require approximations, such as linearising the functions, e.g. in the Extended Kalman Filter (EKF) (see 1.4.1), or deterministic sampling, e.g. in the Unscented Kalman Filter (UKF) (see 1.4.2) to approximate a non-Gaussian density by a Gaussian (Julier and Uhlmann, 2004). Such approximations make the limiting implicit assumption that the true densities are uni-modal. Filters based on these approximations often severely under-perform when true densities are multi-modal. Hence, multi-modal approaches are frequently needed.

For representing multi-modal, non-Gaussian densities particle filters are a standard approach (see 1.4.3). They are computationally demanding since they often require a large number of particles for good performance, e.g. due to the curse of dimensionality. An insufficient number of particles may fail to capture the tails of the density and lead to degenerate solutions. In practice, we have to compromise between the deterministic and fast (UKF/EKF) or the computationally demanding and more accurate Monte Carlo methods (Doucet et al., 2001).

An ideal filter for a non-linear system should allow for multi-modal approximations, and at the same time its approximations should be consistent to avoid degenerate solutions. In this report, we propose a filtering method (see 2.1.3) that approximates a non-Gaussian density by a Gaussian mixture model (GMM). A GMM allows modelling multi-modality as well as representing any density with arbitrary accuracy given a sufficiently large number of Gaussians, see (Anderson and Moore, 2005), Section 8.4, for a proof. Intuitively we can imagine a extreme case of infinite Gaussians with variances tending to zero forming a train of Dirac deltas, that in asymptotically can approximate any function with arbitrary precision. The GMM presents an elegant deterministic filtering solution in the form of the Gaussian Sum filter (Alspach and Sorenson, 1972).

The Gaussian Sum filter (GSUM-F) was proposed as a solution to estimation problems with non-Gaussian noise or prior densities. The GSUM-F relies on linear dynamics and the assumption that the parameters of the Gaussian mixture approximation to the non-Gaussian noise or prior densities are known a priori. This linearity assumption can be relaxed, e.g. by linearisation (EKF GSUM-F) (Anderson and Moore, 2005) or deterministic sampling (UKF GSUM-F) (Luo et al., 2010), but both solutions still require a priori knowledge of the GMM parameters. If, however, the prior and noise densities are Gaussian, the UKF GSUM-F and EKF GSUM-F are reduced to the standard UKF and EKF, i.e. they become uni-modal filters. To account for a possible uni-modal to multi-modal transition in a non-linear system, we need to solve two problems: the propagation of the uncertainty and the parameter estimation of the GMM approximation. Kotecha and Duric (Kotecha and Djuric, 2003), proposed random sampling for uncertainty propagation and Expectation-Maximization (EM) to estimate the GMM parameters. In this report, we propose to propagate uncertainty deterministically using the Unscented Transform, which also allows for a closed-form expression of the GMM parameters.

In our proposed multi-modal filter (Kamthe et al., 2013) we present an approximate solution to address the possible uni-modal to multi-modal transition in a non-linear system by a Gaussian mixture models and derive closed form expressions to estimate Gaussian mixture model parameters, for details see Section 2.1.3.

While filtering is an online estimation, i.e. our current state estimate is based on all the observations up to current time index, we can look back and correct our previous state estimates based on new observations, the filter to correct estimates based on future observations is called 'Fixed lag Smoother' or just 'smoother'. The smoother also needs to take into account the non-linear transition function. We propose a novel smoother based on multi-modal filter in the Section 2.2. To the best of our knowledge the smoothing algorithm for non-linear estimation based on Gaussian is not available. The other Gaussian mixture based smoothers in literature are the " Two-filter smoother" based on the backward information filter (Kitagawa, 1994), the "Gaussian Mixture smoother" (Vo et al., 2012) based on the β recursion and the "Expectation Correction" smoother (Barber, 2006) based on approximating a Gaussian integral by evaluating the integral at mean of the Gaussian uncertainty.

1.1 Scope of the Work and Contributions

As a testimonial to the importance of the non-linear estimation and time series models many standard texts are available, and in particular excellent description of filtering and state estimation problems are covered in texts (Kailath et al., 2000) and (Bar-Shalom et al., 2004).

We include standard Gaussian filters and particle filters as state of the art non-linear estimators in introductory chapter. Non-linear filters can be broadly classified as described in 1.2. We introduce the only the standard algorithms for each theme, e.g. Determinisitic unimodal filters (Gaussian beliefs) also include Cubature Kalman filters (Arasaratnam and Haykin, 2009), Gauss Hermite Quadrature filters, etc.

1.1.1 Scope

In this work we restrict ourselves to Gaussian priors, Gaussian Noise densities and use standard benchmarking tools like Univariate Non-linear Growth Model (UNGM) to test new algorithms proposed in this work. The Gaussian sum filter and Multi-modal filters can handle non-Gausian densities and, hence, will outperform uni-modal non-linear filters in non-Gaussian noise density scenarios. However, we avoid non-Gaussian priors to demonstrate that the proposed algorithm can predict and handle non-Gaussian densities generated due to non-linearity of systems. The proposed mulit-modal filter will outperform uni-modal filters under standard testing conditions for non-Gaussian priors [2], hence, for fair comparison with uni-modal filters we only consider a Gaussian noise in the system.

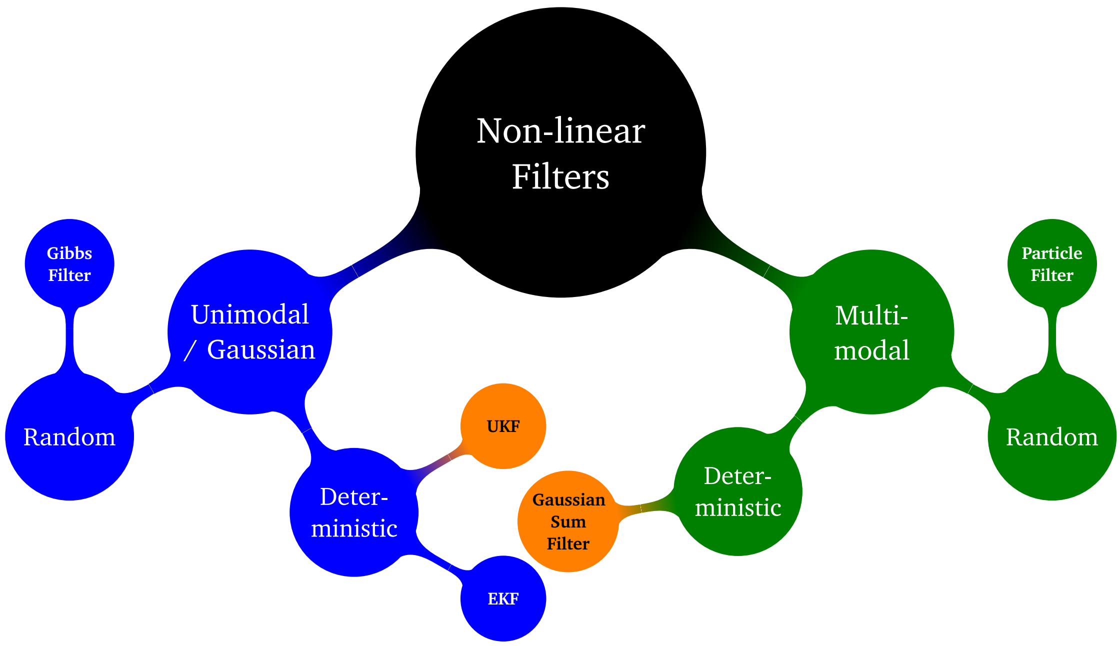


Figure 1.2.: Non-linear Filters: The classification of non-linear filters is abstract and its main purpose to define scope of this work in context of state of the art literature, e.g. all leafs are non-linear filters and described in this report. Orange circles represent the standard algorithms on which we establish our contributions.

[2] Multi-modal filter is based on (Alspach and Sorensen, 1972)

Assumptions and Scope

Unless stated otherwise, the following assumptions are made throughout the report.

- **Non-linearity:** The system function f and measurement function h are non-linear and assumed to be known and continuous.
- **Prior:** We assume a Gaussian prior on initial state variables
- **Noise:** Noise density is assumed to be Gaussian and we assume standard normal distribution through out our discussion and examples to maintain consistency.

The following points are **NOT** covered in this report.

- A thorough comparison with Markov-Chain-Monte-Carlo (MCMC) methods.
- Comparison of proposed smoother with state of the art Gaussian smoothers.

We, however, include a comparison with Gibbs smoother (Deisenroth and Ohlsson, 2011) and leave thorough comparisons to future work.

1.1.2 Contributions

The main contributions of this thesis are enlisted below.

1. We present closed-form expressions for estimating the parameters of the Gaussian mixture model to approximate multi-modal predictive density of a non-linear dynamical system, see Section 2.1.2.

2. Multi-Modal-Filter (M-MF) (Kamthe et al., 2013), introduced in Section 2.1 is a multi-modal approach to filtering in non-linear dynamical systems, where all densities are represented by Gaussian mixtures.

3. We propose a novel multi-modal smoothing pass for non-linear dynamical systems, the result from Multi-Modal Filter is used to obtain a Gaussian Mixture based multi-modal smoother, see Section 2.2. A smoothing pass based on Gaussian sum approximations is not available in literature to the best of our knowledge, and this algorithm is the most significant contribution of the thesis. Note: Gaussian Sum Approximations were first introduced in (Alspach and Sorensen, 1972) and form the basis of Gaussian sum filter (Anderson and Moore, 2005).

The Gaussian mixture representation provides flexibility to approximate complex predictive densities along with ability to use closed form expressions for filtering and smoothing.

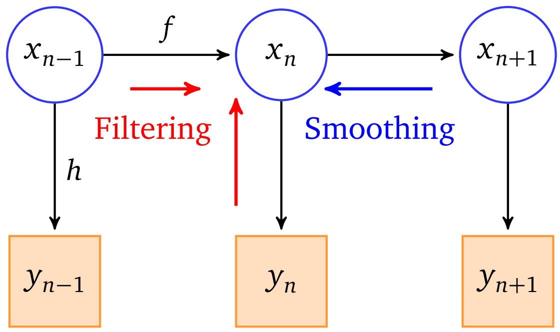


Figure 1.3.: State Space Model (SSM) of a dynamical system. Circles represent the hidden state variables, squares represent observed variables and subscripts denote time indices. State transition function f and measurement function h are assumed known and non-linear. Arrows indicate information flow in filtering and smoothing and both estimators are recursive, i.e. in a typical SSM model the above sequence repeats itself in both directions (left and right).

We now describe the generic framework of inference in state space models and we introduce standard terminology in the context of tennis ball tracking. Readers familiar with state-space models can skip to Bayesian filters in Section 1.2.1.

The mathematics or equations to *'predict'* the future state values and *'correct'* our estimate of the state based on the new measurements is called filtering. When the system is non-stationary and noise is Gaussian we can use Kalman Filter (Kalman, 1960) to obtain optimal estimates (Anderson and Moore, 2005).

The state-space models are powerful tools to represent many complex problems elegantly, e.g. whole ball tracking system can be neatly and compactly represented by the state-space diagram shown in Figure 1.3. The state-space representation is simple enough to understand intuitively and yet it can represent almost all the estimation problems of practical significance (Thrun et al., 2005).

In the diagram the states are represented by circles and measurements by squares . The circles are called the latent variables as we can ascertain their existence only through our observations, the orange squares in the Figure 1.3. The velocity of tennis ball for example, is a latent variable and we can estimate it based on our observations only.

To facilitate a mathematical formulation of system we assume Markov property, i.e. the next state is based entirely on the current state and is independent of all the other states. The Markov assumption allows us to write recursive estimation algorithms.

Throughout this thesis we represent state of the system by variable x and use subscript notation x_n to denote values of the state vector at discrete time step index n. To model sequential data in time series we use SSM as described above. The state transition function f and measurement(observation) function h are assumed to be known. The variable $Y_j = \{y_1, \ldots, y_j\}$ represents all the observations up to time step j. The model can be represented as in Figure 1.3.

1.2.1 Filtering

In a recursive Bayesian filter, the current belief or the state distribution $p(x_n)$ is calculated from the previous state distribution $p(x_{n-1})$ at a discrete time index $n-1$. We initialise the recursive filter with the state x_0 and a distribution $p(x_0)$. As our state is latent, the current belief is conditioned on observations up until the current observation, i.e. the current belief is $p(x_{n-1}|Y_{n-1})$. Filtering is defined as estimating $p(x_n|Y_n)$ given new observation y_n and can be divided in two steps, the time update and the measurement update. We typically, alternate between these two steps, i.e. we predict the state (time update) and then correct the state estimate (measurement update) and use the new corrected state to predict next state and so on.

Time Update

In the time update we move one step horizontally along the chain (horizontal red arrow) in the Figure 1.3. This can be represented as estimating $p(x_n|Y_{n-1})$ from $p(x_{n-1}|Y_{n-1})$. The time update is defined by the integral

$$p(x_n|Y_{n-1}) = \int p(x_n|x_{n-1})p(x_{n-1}|Y_{n-1})\,\mathrm{d}x_{n-1}. \tag{1.1}$$

This integral cannot be solved in closed form for non-linear systems. All non-linear filters approximate this integral. Particle filters replace the integral by sum and probabilities by random samples.

Measurement Update

In the measurement update we incorporate the current observation y_n and estimate the posterior density $p(x_n|Y_n)$. This can be interpreted as moving information from observation to correct the time update prediction according to the current measurement (vertical red arrow). We use Baye's theorem to write

$$p(x_n|Y_n) = \frac{p(y_n|x_n)p(x_n|Y_{n-1})}{p(y_n|Y_{n-1})}, \tag{1.2}$$

where the marginal density $p(y_n|Y_{n-1})$ is obtained by $\int p(y_n|x_n)p(x_n|Y_{n-1})\,\mathrm{d}x_n$.

1.2.2 Smoothing

In smoothing, we wish to estimate the distribution of the state x_n given all observations Y_N. We assume that all the state predictive densities $p(x_n|Y_n)$ are available. We estimate $p(x_n|Y_n)$ recursively as (Kitagawa, 1994)

$$p(x_n|Y_N) = p(x_n|Y_n)\int \frac{p(x_{n+1}|Y_N)p(x_{n+1}|x_n)}{p(x_{n+1}|Y_n)}\mathrm{d}x_{n+1}. \tag{1.3}$$

In the Figure 1.3, this can be interpreted as moving from right to left.

Tennis Ball Tracking

We now consider Bayesian filtering in the ball tracking context.

Time Update:
: We estimate the next state of the ball. Will it move in a straight line? Will it hit the ground (in next time step)? Was there any spin?

Measurement Update:
: We 'correct' our estimate to update our current state, e.g. it hit the ground, there was a top spin imparted on it, etc.

Smoothing Update:
: Smoothing allows us to correct the whole trajectory based on the final observations, i.e. now we know that there was top spin, does it change our estimate of the state of the ball at the very beginning (Ideally, it should, if the spin imparted was our latent variable and now we know more about, the type and the amount, of spin based on the final result).

So if we know how the system works and have high speed cameras why do we need integrals and probabilities? Our observations are noisy or uncertain (capturing 3D world with 2D sensors), even with stereoscopic vision we can be tricked (optical illusions). In the Hawk-Eye tracking system the ball and its environment are uncertain, e.g. humidity and amount of spin imparted decide how the ball moves in air. So to deal with uncertainties we make assumptions, and naturally the accuracy of estimation depends on how close our assumptions are to the reality. In the following we establish a standard approximation used in Bayesian filtering and smoothing.

1.3 Linear Gaussian Filter

The Bayesian filtering as defined by the Equations (1.1) to (1.3), can be solved in closed form with Linear Gaussian systems (Thrun et al., 2005). The derivation of the Kalman filter from Bayesian filter is well known and we do not attempt to rewrite it. Readers can find Kalman filter derivation based on Wiener filter in (Kailath et al., 2000), a derivation based on Bayesian filter approach is given in (Thrun et al., 2005).

The Kalman filter exploits Gaussian conditioning property, i.e. if we have a jointly Gaussian distribution we can obtain conditional density in a closed form. We assume Gaussian uncertainties and approximate densities or beliefs as Gaussians. The filtering based on the Gaussian assumption is called Gaussian filtering.

1.3.1 The Kalman Filter

We consider a dynamical system described by the following equations

$$x_n = f(x_{n-1}) + w_n, \qquad w_n \sim \mathcal{N}(0, Q), \tag{1.4}$$
$$y_n = h(x_n) + v_n, \qquad v_n \sim \mathcal{N}(0, R), \tag{1.5}$$

where f and h are the (non-linear) transition and measurement functions, respectively. The noise processes w_n and v_n are i.i.d. zero mean Gaussian with covariances Q and R, respectively. We denote the D dimensional state by x_n, and y_n is the E dimensional observation.

In Gaussian filtering we assume that all the known densities , system and measurement noise and prior density are Gaussian. For a linear system this enables us to use linear Gaussian conditioning and all the equations (1.1) to (1.3) can be derived in closed form.

We assume that our state and measurement variables are jointly Gaussian to exploit the Gaussian conditioning property. The assumption reduces Kalman filtering to estimating the auto-covariance and cross-covariance matrices (Deisenroth and Ohlsson, 2011).

The joint Gaussian matrix for a latent state x_n and measurement variable y_n can be written as

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} \sim \mathcal{N}\left(\begin{matrix} x_n \\ y_n \end{matrix} \;\middle|\; \begin{pmatrix} \mu_n \\ \mu_n^y \end{pmatrix}, \begin{pmatrix} \Sigma_n^{x,x} & \Gamma_n^{x,y} \\ \Gamma_n^{x,y\,T} & \Sigma_n^{y,y} \end{pmatrix}\right) \tag{1.6}$$

and for state variables at time index n and $n-1$ as,

$$\begin{pmatrix} x_{n-1} \\ x_n \end{pmatrix} \sim \mathcal{N}\left(\begin{matrix} x_{n-1} \\ x_n \end{matrix} \;\middle|\; \begin{pmatrix} \mu_{n-1} \\ \mu_n \end{pmatrix}, \begin{pmatrix} \Sigma_{n-1}^{x,x} & \Gamma_n^{x,x} \\ \Gamma_n^{x,x\,T} & \Sigma_n^{x,x} \end{pmatrix}\right) \tag{1.7}$$

Algorithm 1 Kalman Filter: Algorithm illustrates Kalman filtering for Linear Gaussian systems. Subscripts define time index, superscripts denote latent x or observed y variable, μ, Σ, and Γ are mean, covariance and cross covariance respectively of a Gaussian (Linear systems) or approximated as Gaussian (Non-linear systems) beliefs

function KalmanFilter($f, h, \mu_{n-1}, \Sigma_{n-1}, y_n, Q, R$)
  $\widehat{\mu_n} = f(\mu_{n-1})$
  Estimate $\Sigma_n^{x,x}$ using f and Σ_{n-1}
  $\widehat{\Sigma_n} = \Sigma_n^{x,x} + Q$ ▷ Time update
  Estimate $\Sigma_n^{y,y}$ and $\Gamma_n^{x,y}$ using h and $\widehat{\Sigma_n}$
  $K_n = \Gamma_n^{x,y}\left(\Sigma_n^{y,y} + R\right)^{-1}$ ▷ Kalman Gain K_n
  $\mu_n = \widehat{\mu_n} + K_n\left(y_n - h(\widehat{\mu_n})\right)$
  $\Sigma_n = \Sigma_{n-1} - (\Gamma_n^{x,y})(\Sigma_n^{y,y})(\Gamma_n^{x,y\,T})$ ▷ Measurement Update
  return μ_n, Σ_n
end function

We describe Kalman filter algorithm based on the joint Gaussian assumption in Algorithm 1. For the linear dynamical system with system gain matrix F_n and measurement matrix H_n the covariances are given by

$$\begin{aligned}
\Sigma_n^{x,x} &= F_n \Sigma_n F_n^T, \\
\Sigma_n^{y,y} &= H_n \Sigma_n H_n^T, \\
\Gamma_n^{x,x} &= \Sigma_n F_n^T, \\
\Gamma_n^{x,y} &= \Sigma_n H_n^T.
\end{aligned} \tag{1.8}$$

For a non-linear system with a Gaussian assumption (implicit linearisation), we approximate these covariances matrices using different methods such as the Unscented Transform or Gibbs sampling (Deisenroth and Ohlsson, 2011).

1.3.2 The Rauch-Tung-Striebel (RTS) Smoother

In smoothing we move backwards to update our estimate of trajectory. Below we describe a well-known backward pass called Rauch-Tung-Striebel (RTS) smoother. The derivation of the backward pass below is similar to the backward pass presented in (Särkkä, 2008).

We describe below a procedure to evaluate integral (1.3).

1. We reproduce the Equation (1.3)

$$p(x_n|Y_N) = p(x_n|Y_n) \int \frac{p(x_{n+1}|Y_N)p(x_{n+1}|x_n)}{p(x_{n+1}|Y_n)} \mathrm{d}x_{n+1}. \tag{1.9a}$$

2. We push the filtering result $p(x_n|Y_n)$ inside the the integral as it a constant w.r.t. integration variable x_{n+1} to obtain

$$p(x_n|Y_N) = \int \underbrace{p(x_{n+1}|x_n)p(x_n|Y_n)}_{\text{joint distribution of } x_n \text{ and } x_{n+1}} \frac{p(x_{n+1}|Y_N)}{p(x_{n+1}|Y_n)} \, \mathrm{d}x_{n+1}. \tag{1.9b}$$

3. We can obtain conditional distribution of x_n given x_{n+1} and Y_n using the joint distribution $p(x_n, x_{n+1}|Y_n)$. This conditional can be achieved by mere reformulation of terms in step above

$$p(x_n|Y_N) = \int \underbrace{\frac{p(x_n, x_{n+1}|Y_n)}{p(x_{n+1}|Y_n)}}_{\text{Conditional distribution of } x_n} p(x_{n+1}|Y_N) \, \mathrm{d}x_{n+1}. \tag{1.9c}$$

4. Exploiting the Markov property, we can write $p(x_n|x_{n+1}, Y_N) = p(x_n|x_{n+1}, Y_n)$ and, hence,

$$p(x_n|Y_N) = \int p(x_n|x_{n+1}, Y_N)p(x_{n+1}|Y_N) \, \mathrm{d}x_{n+1}. \tag{1.9d}$$

Algorithm 2 RTS Smoother: The algorithm describes RTS smoother for linear Gaussian systems. Subscripts define time index, superscripts denote latent x or observed y variable, μ, Σ, and Γ are mean, covariance and cross covariance respectively of a Gaussian (Linear systems) or approximated as Gaussian (Non-linear systems) beliefs. Superscript 's' is used to denote final smoothed result, i.e. we define smoothed mean as μ_n^s and filtered mean as μ_n
.

function RTSSMOOTHER($\mu_n, \Sigma_n, \mu_{n+1}, \Sigma_{n+1}, Q$)
    $\widehat{\mu}_{n+1} = f(\mu_n)$
    $\widehat{\Sigma}_{n+1} = F_n \Sigma_{n-1} F_n(\mu_n)^T + Q$
    $D_n = \Sigma_n F_n \widehat{\Sigma}_{n+1}^{-1}$ ▷ Smoother Gain D_n
    $\mu_n^s = \mu_n + D_n \left(\mu_{n+1} - \widehat{\mu}_{n+1}\right)$
    $\Sigma_n^s = \Sigma_n + D_n \left(\Sigma_{n+1} - \widehat{\Sigma}_{n+1}\right) D_n^T$
    return μ_n^s, Σ_n^s
end function

5. If we assume

$$p(x_n|x_{n+1}, Y_N) \sim \mathcal{N}(x_n|\mu_n^{'}, \Sigma_n^{'}) \quad \text{and}$$
$$p(x_{n+1}|Y_N) \sim \mathcal{N}(x_n|\mu_{n+1}^s, \Sigma_{n+1}^s),$$

then the distribution $p(x_n|Y_N)$ is also Gaussian, i.e.

$$p(x_n|x_{n+1}, Y_N) \sim \mathcal{N}(x_n|\mu_n^s, \Sigma_n^s)$$

Based on Step 5 above, we can use Gaussian conditioning to write smoothing algorithm as Algorithm 2,

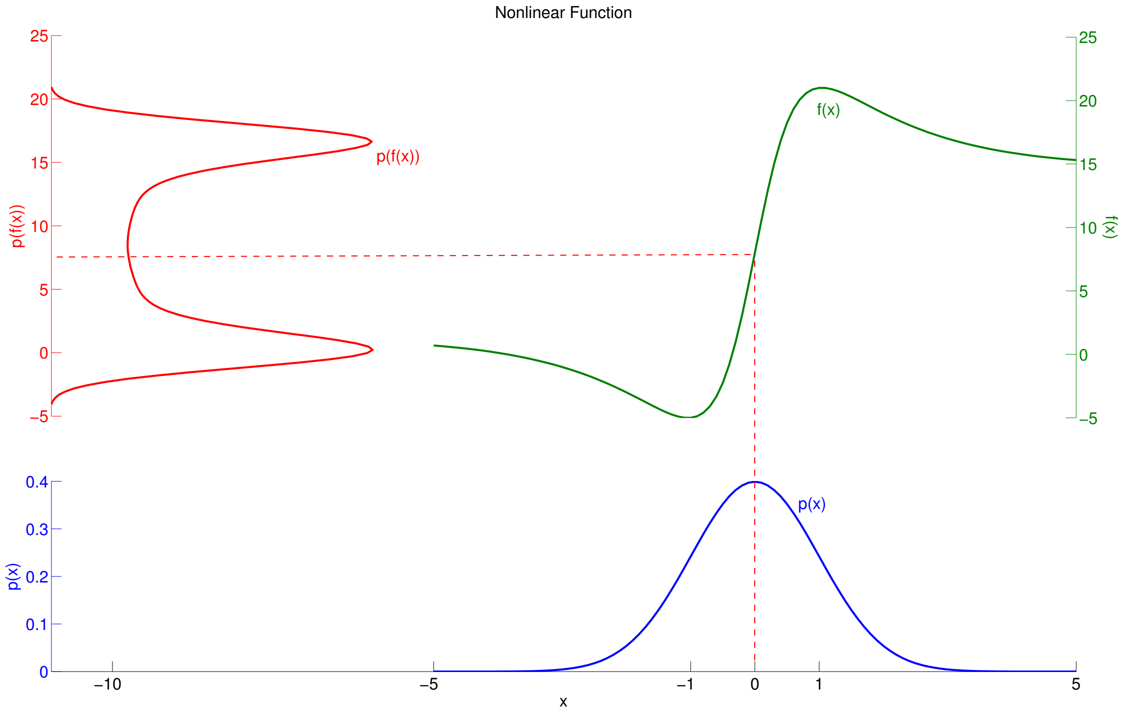


Figure 1.4.: Example of Multi-modal predictive density in UNGM with standard normal distribution as prior. The blue curve is standard normal prior and the green curve represents a non-linear function. The predictive density for a non-linear mapping of standard normal distribution is shown by red curve. The red curve was generated using kernel smoothing density estimator using 10^5 samples. The samples are generated from the Gaussian distribution shown in blue and then mapped by non-linear function f (in green).

1.4 Nonlinear Filters

The Kalman filter and RTS smoother allow us to recursively calculate state estimates and all relations are given in closed form, however, they make assumption that the system is linear. Most applications of real life examples are nonlinear in nature.

To understand it in our ball tracking system, our observation function, the video cameras implicitly use non-linear transformations to measure position of the ball, e.g. Cartesian to polar, 3 dimensional world to 2 dimensional observation. We cannot use Kalman Filter in its form described in Section 1.3.1 unless we use approximations to estimate the covariance matrices in Equations (1.6) and (1.7). The approximations are needed due to the following challenges in non-linear dynamical systems.

1. Predictive distributions or the integrals used in our Bayesian framework do not yield closed form solutions. Approximations are needed.

2. Non-linear mapping results in non-Gaussian distributions, i.e. even if we were able to somehow evaluate the integrals, we cannot propagate the results forward due to the lack of parametric form. As an illustration, consider a hypothetical system that starts as Gaussian

and undergoes a nonlinear function map as shown in Figure 1.4. The red curve on left describes non-Gaussian density hence we cannot use Kalman filter or RTS smoother.

Most non-linear estimators solve both challenges simultaneously i.e by linearising a non-linear system. The process is described in the following sections. We make the above distinction to illustrate our approach to tackle the problem.

We treat above challenges individually, i.e.

1. Provide a Bayesian framework to estimate predictive density
2. The density is parametric hence we can use recursive filters like Gaussian sum filters and yet it is not necessarily linearisation as we allow a non-Gaussian predictive density to a Gaussian prior. The Figure 1.4 shows approximation for the same non-linear function we described in Figure 2.2.

If we approximate the predictive density by a Gaussian we implicitly linearise the system. This implicit linearisation can lead to large approximation errors, e.g. see Figure 1.4. In Figure 1.4, any Gaussian approximation to the red curve (predictive density) would result in large errors as a single Gaussian cannot capture both modes simultaneously.

Linear Approximation

As described above in the linear approximation we approximate all the beliefs or state distributions by a Gaussian density. The Gaussian assumption allows us to use property of Gaussian conditioning a property that enables us to write state estimation in closed form.

So, if we assume $p(x_n) \sim \mathcal{N}(x_n|\mu_n, \Sigma_n)$ and $p(x_{n-1}) \sim \mathcal{N}(x_{n-1}|\mu_{n-1}, \Sigma_{n-1})$, then our linear approximation implies

$$\begin{pmatrix} x_n \\ x_{n-1} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} x_n \\ x_{n-1} \end{pmatrix} \middle| \begin{pmatrix} \mu_n \\ \mu_{n-1} \end{pmatrix}, \begin{pmatrix} \Sigma_n & \Gamma_n \\ \Gamma_n^T & \Sigma_{n-1} \end{pmatrix} \right). \tag{1.10}$$

With the known transition function f we can evaluate the mean of the predictive Gaussian as $\mu_n = f(\mu_{n-1})$. In the following we describe standard methods classified based on the way we approximate the predictive Covariances Σ and Γ. A unified view on the linear Gaussian systems is given in (Deisenroth and Ohlsson, 2011). We give brief overview of some non-linear estimators that use the linear Gaussian assumption.

Extended Kalman Filter: Linearise the function using Taylor series expansion, see Section 1.4.1.

Unscented Kalman Filter: Use deterministic sampling to estimate the Covariances, see Section 1.4.2.

Gibbs Filter: Random sampling to estimate the mean and the Covariances, see Section 1.4.3.

The list of algorithms is meant to be illustrative and not exhaustive. Each type described above demonstrates a principle or concept used to approximate to tackle intractability inherent in non-linear filtering systems (Deisenroth and Ohlsson, 2011). Several ad-hoc methods are available in literature.

1.4.1 Extended Kalman Filter

As described in Section 1.4, we linearise the system and measurement functions. We use the same model as described by system of Equations (1.4). With our linear approximation model (1.10), the parameters of predictive Gaussian are calculated as

$$
\begin{aligned}
\mu_n &= f(\mu_{n-1}), \\
\Sigma_n^{x,x} &= F_{n-1}(\mu_{n-1})(\Sigma_{n-1})F_{n-1}^T(\mu_{n-1}), \\
\Sigma_n^{y,y} &= H_n(\mu_n)(\Sigma_n)H_n^T(\mu_n), \\
\Gamma_n^{x,y} &= \Sigma_{n-1}H_{n-1}^T(\mu_{n-1}), \\
\Gamma_n^{x,x} &= \Sigma_{n-1}F_{n-1}^T(\mu_{n-1}),
\end{aligned}
\tag{1.11}
$$

where F_{n-1} is the Jacobian matrix of f obtained by

$$
\left[F_n(\mu_n)\right]_{k,k'} = \left.\frac{\partial\ f_k(x)}{\partial_{k'} x}\right|_{x=\mu_n}. \tag{1.12}
$$

where k and k' denote row and column of a matrix, similarly we have

$$
\left[H_n(\mu_n)\right]_{k,k'} = \left.\frac{\partial\ h_k(y)}{\partial_{k'} y}\right|_{x=\mu_n}. \tag{1.13}
$$

The matrices in (1.11) can be used in Algorithm 1 to obtain Extended Kalman Filter (EKF).

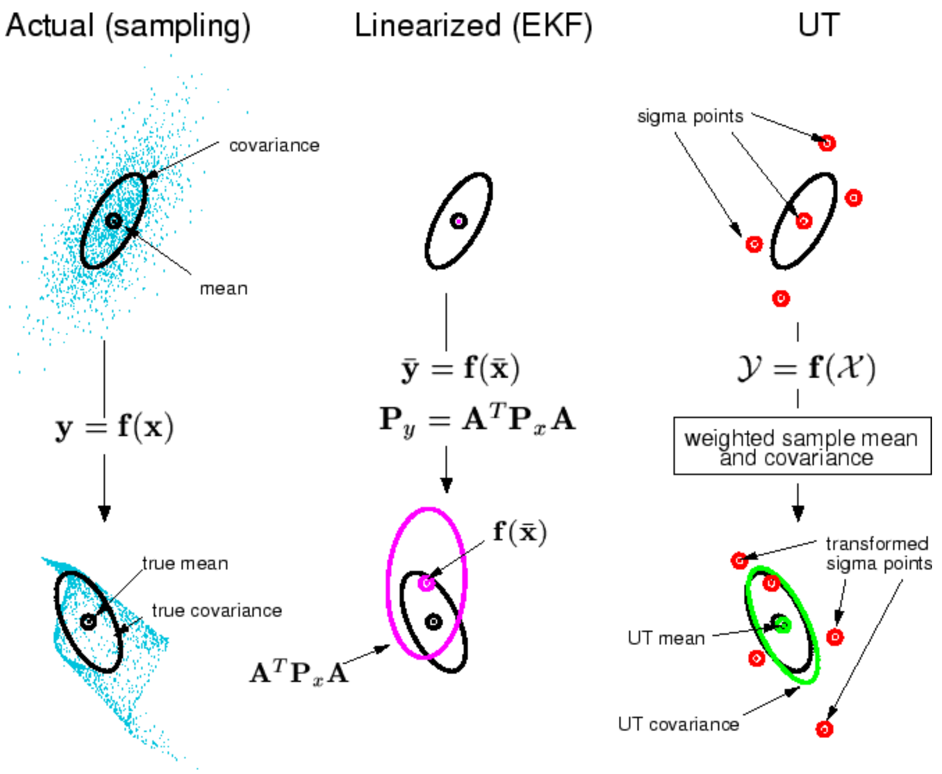


Figure 1.5.: Example of the UT for mean and covariance propagation. a) actual, b) first-order linearization (EKF), c) UT. The Figure is reproduced from (Wan and van der Merwe, 2000)

1.4.2 Unscented Kalman Filter

The unscented Kalman filter is based on the Unscented Transform (UT) (Julier and Uhlmann, 2004) where we use deterministic sampling to estimate the covariance matrices Σ_n and Γ_n. Similar to EKF we need only estimate the covariance matrices (Deisenroth and Ohlsson, 2011) to obtain Unscented Kalman filter.

The matrices in (1.14) can be used in Algorithm 1 to obtain Unscented Kalman Filter (UKF). To illustrate the UT consider a variable $x_{n-1} \sim \mathcal{N}(x_{n-1}|\mu_{n-1}, \Sigma_{n-1})$ and we want to estimate the distribution of $p(x_n)$ gievn that $x_n = f(x_{n-1})$ and f is any non-linear function. We can divide the transform in the following steps,

1. Calculate sigma points X_n of distribution $\mathcal{N}(x_{n-1}|\mu_{n-1}, \Sigma_{n-1})$.

2. Propagate the sigma points through non-linear function f.

3. Estimate transformed mean and covariances based on the transformed sigma points. [3]

[3] The Y_n in (1.14f) and (1.14g) is a matrix and not a vector of observations.

$$\chi_{n-1}^{[0]} = \mu_{n-1}, \tag{1.14a}$$
$$\chi_{n-1}^{[j]} = \mu_{n-1} + (\sqrt{\Sigma_{n-1}})^j, \tag{1.14b}$$
$$\chi_{n-1}^{[j+D]} = \mu_{n-1} - (\sqrt{\Sigma_{n-1}})^j, \tag{1.14c}$$

where $(\sqrt{\Sigma_{n-1}})^j$ represents j th row of the Cholesky factor of Σ_{n-1}. The points $\mathcal{X}_{n-1}$ are called sigma points as they are spread around mean at a distance of one sigma

$$\mu_n = X_n w_\mu, \tag{1.14d}$$
$$\Sigma_n^{x,x} = X_n W X_n^T, \tag{1.14e}$$
$$\Sigma_n^{y,y} = Y_n W Y_n^T, \tag{1.14f}$$
$$\Gamma_n^{y,x} = Y_n W X_n^T, \tag{1.14g}$$

where $\mathcal{X}_{n-1}$ is a sigma point matrix and $\sqrt{\Sigma_{n-1}}$ is a lower triangular matrix of the Cholesky factorisation, function $f(.)$ is applied to each column of argument matrix. The constant is obtained by $c = \alpha^2(n+\kappa)$ where α and κ are constants of UT. The vector w_μ and matrix W are defined as

$$w_\mu = \begin{bmatrix} W_m^0 & \dots & W_m^{2n} \end{bmatrix}^T, \tag{1.14h}$$
$$W = \Big[(I - [w_\mu \dots w_\mu])\Big]\Big[diag(W_c^0 \dots W_c^{2n})\Big]\Big[(I - [w_\mu \dots w_\mu])\Big]^T. \tag{1.14i}$$
$$\tag{1.14j}$$

Where the weights are obtained by

$$\begin{aligned} W_m^0 &= \lambda/(n+\lambda), \\ W_c^0 &= \lambda/(n+\lambda) + (1-\alpha^2+\beta), \\ W_m^i &= 1/\{2(n+\lambda)\}, \qquad i = 1, \dots, 2n., \\ W_c^i &= 1/\{2(n+\lambda)\}, \qquad i = 1, \dots, 2n. \end{aligned} \tag{1.14k}$$

with λ defined as

$$\lambda = \alpha^2(n+\kappa) - n. \tag{1.14l}$$

The hyperparameters λ, α and κ control the spread of the sigma points $\mathcal{X}$ (?)

Algorithm 3 Particle Filter: The following algorithm describes a generic particle filter algorithm, adapted from (Thrun et al., 2005). The $\mathcal{X}$ represents samples, subscript denotes time index. The system function and the measurement functions are given by f and h respectively. The algorithm returns particles $\mathcal{X}_n$ that collectively represent filtered estimate of system at time n.

function PARTICLE FILTER($\mathcal{X}_{n-1}, f, h, y_n$)
 $\widehat{\mathcal{X}_n} = \mathcal{X}_n = \emptyset$
 for m=1 to M **do**
  sample $x_n^{[m]} \sim f(x_{n-1}^{[m]}) \sim p(x_n|x_{n-1}^{[m]})$
  $w_n^{[m]} = p(y_n \mid h(x_n^{[m]}))$ ▷ Calculate weight (importance) of particle
  $\widehat{\mathcal{X}_n} = \widehat{\mathcal{X}_n} + \left\langle x_n^{[m]}, w_n^{[m]} \right\rangle$
 end for
 for m=1 to M **do**
  sample i with probability $\propto w_n^{[i]}$ ▷ Importance re-sampling
  add $x_n^{[i]}$ to $\mathcal{X}_n$
 end for
 return $\mathcal{X}_n$
end function

1.4.3 Particle Filters

Particle filters are non parametric implementation of Bayesian filter. In particle filters we represent our beliefs or probabilities densities by samples drawn from these densities. See (Thrun et al., 2005) for detailed overview on particle filters in terms of Bayesian filtering.

As an illustrative example consider a system with prior probability,

$$p(x_0|Y_0) = \mathcal{N}(x_0|\mu_0, \Sigma_0).$$

Subsequently, we represent $p(x_0|Y_0)$ by set $\mathcal{X}$ of randomly drawn samples from $\mathcal{N}(x_0|\mu_0, \Sigma_0)$, i.e.

$$\mathcal{X}_0^{[M]} \sim p(x_0|Y_0) = \mathcal{N}(x_0|\mu_0, \Sigma_0)$$

where M represents the number of particles.

For the time update we can use the function f on set of particles $\mathcal{X}_{n-1}^{[M]}$, i.e.

$$\mathcal{X}_n^{[M]} \sim p(x_n|Y_{n-1}) \sim f(\mathcal{X}_{n-1}^{[M]})$$

Generic particle filter is described in algorithm 3 which is adapted from (Thrun et al., 2005) for a thorough overview of particle filters and Markov Chain Monte Carlo (MCMC) methods refer (Doucet et al., 2001).

Gibbs Filter

Unlike generic particle filters described in Section 1.4.3 Gibbs filter is parametric. The state densities $p(x_n|Y_n)$ are approximated by a parametric distribution and we use random sampling (Gibbs sampler) to estimate parameters of these approximate density.

In (Deisenroth and Ohlsson, 2011), the authors use the Gaussian assumption for predictive densities and the state covariance and cross covariance matrices are calculated using Gibbs sampler. In this work we use Gibbs filter described in (Deisenroth and Ohlsson, 2011) to compare performance of the proposed filter.

1.4.4 Gaussian Sum Filter

The non-linear filters we discussed in preceding sections all share a common philosophy, approximate a intractable predictive distribution by a Gaussian distribution. Gaussian approximation also implicitly imposes a unimodal belief on our state estimate, however, as the name of the report suggests we are interested in multi-modal beliefs. If we express our multi-modal beliefs as Gaussian mixtures then we can use Gaussian Sum Filter proposed by (Alspach and Sorenson, 1972). We would like to stress that we view Gaussian sum filter as an algorithm to Kalman Filter like approximations when beliefs are expressed as Gaussian Mixture. The difference is subtle and we give following examples to support our view.

1. Gaussian sum filter can be used for non-linear systems if and only if we express approximate beliefs as Gaussian Mixtures. Gaussian sum filter has no mechanism to deal with non-linearities. Various ways to approximate functions using optimisation are discussed by authors in (Alspach and Sorenson, 1972).

2. In Switching Linear Dynamical Systems (SLDS) Gaussian mixtures arise naturally, Gaussian sum filter can be expressed as a Assumed Density Filter [cite Minka, Barber] where the assumed density is Gaussian Mixture.

An excellent discussion and alternative view on usage of Gaussian sum for non-linear filtering is given in Section 8.4 of (Anderson and Moore, 2005). The chapter covers several pages and we recommend it as must read to gain insights in the interpretations of Gaussian Sum Filter. However, the argument “ Gaussian Sum Filter can be used as an approximate inference technique with Gaussian Mixture beliefs ” is the important take away point from this Section.

Estimation Scheme	Mean	Variance	$P(X \leq 0)$
χ^2 distribution	1.0	2.0	0
Particle Filter 1e3 [1e6] particles	0.9599 [0.9998]	1.8569 [2.0020]	0
Extended Kalman Filter	0	$N/A^{\dagger}$	$N/A^{\dagger}$
Unscented Kalman Filter	1.1	4.8400	0.3085
Multi-modal Filter	1.1	1.3	**0.1680**

Table 1.1.: Comparison of various deterministic non-linear approximations: The table shows approximations for standard normal distribution $p(x) = \mathcal{N}(0,1)$ mapped through quadratic non-linearity x^2. True distribution is chi-squared with 1 degree of freedom $p(x^2) = \chi_1^2$. The unscented Kalman filter and multi-modal filter use the same optimal hyper parameters, multi-modal filter in addition uses tuning parameter $\alpha = 1.2$. † The EKF diverges, see Figure 1.6

1.5 Summary of State of the Art Filters and Motivation for Multi-Modal Filter

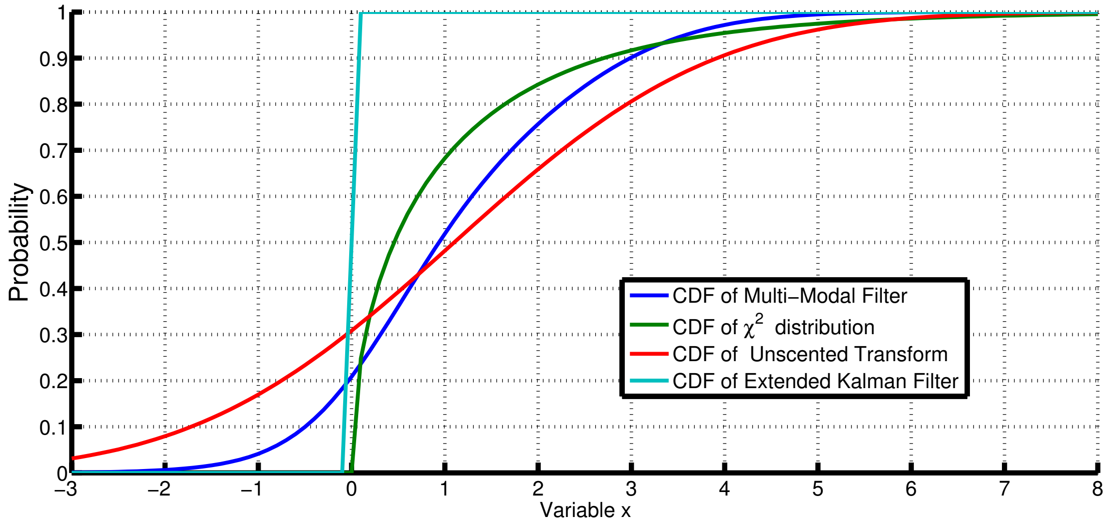
The Extended Kalman filter and Unscented Transform filter are standard state of the art algorithms for deterministic inference with Gaussian beliefs, a similar method for multi-modal beliefs is not available to the best of our knowledge. The success of EKF and UKF is based on the ease of implementation and approximations may result in severe errors even for most trivial problems.

We motivate need for a fast deterministic method with a simple example of one dimensional quadratic map to demonstrate. The Section is motivated by discussions in (Gustafsson and Hendeby, 2012). We use quadratic approximation to test the methods.

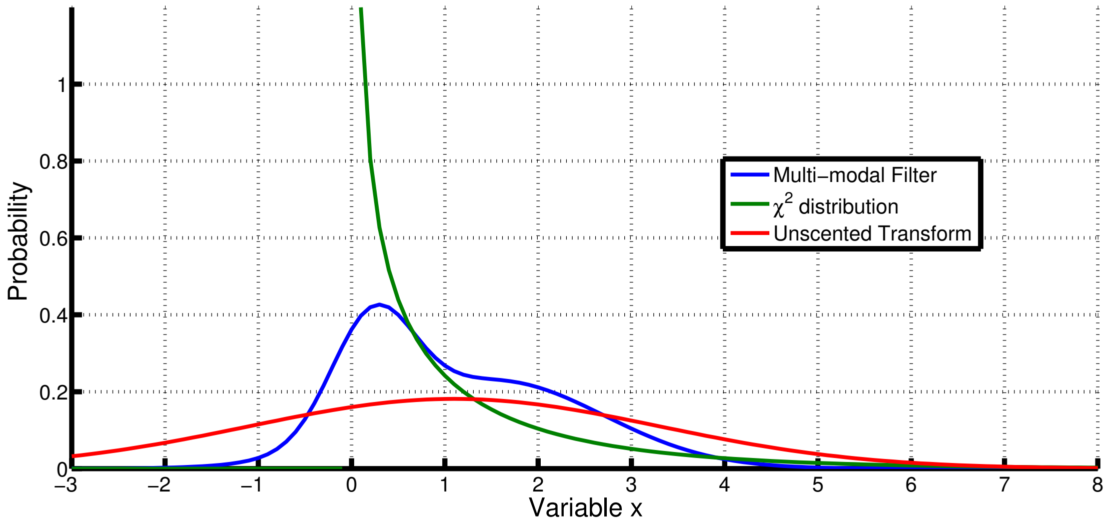
- Large number of real life applications often have quadratic map, e.g. the euclidean distance in tracking problems is a quadratic map.
- Quadratic map is simple enough that we can calculate desired values in closed form.
- The probability distribution is not a Gaussian and we can estimate errors in approximation.

We consider a map $f(x) = x^2$ for a standard Gaussian variable $x \sim \mathcal{N}(x|0,1)$. The $p(f(x))$ is then a χ_n^2 distribution with n degrees of freedom, where n is dimension of the variable x. The plots of approximations for prediction with UKF, EKF, Particle Filter(sampling) and true χ_1^2 distribution are plotted in Figure 1.6.

We discuss this results along with the proposed Multi-modal filter in Chapter 4. Here, the main aim of presenting these results is, both EKF and UKF may result in severe errors, particle filters need substantially large number of particles to get good approximations, an approximation may not match true moments of transformed variable and yet result in less approximation errors, see Figure 1.6 and Table 1.1.



(a) Cumulative Distribution Function



(b) Probability Distribution Function

Figure 1.6.: Comparison of various deterministic non-linear approximations: The figure shows predictive densities for a quadratic transformation (x^2) of the standard normal distribution by non-linear estimation methods. The green curve shows the ideal result, χ^2 distribution with 1 degree of freedom (χ_1^2) (a) The plot shows cumulative distribution of the predicted densities, ideally there should be no mass before zero as shown by the green curve of χ^2 distribution. The EKF concentrates whole mass at $x = 0$ (b) The plot shows predictive densities. Note, the χ_1^2 distribution has discontinuity at $x = 0$

2 Multi-Modal Filter and Smoother

We discussed shortcomings of unimodal approximations in Section 1.5. The ideal filter for a non-linear system should be able to represent multi-modal densities and make consistent estimates. The multi-modal beliefs give the filter additional flexibility to reduce approximation errors. A Gaussian mixture is an ideal approximation for multi-modal beliefs; it is parametric, and we can use Gaussian Sum filter for estimation. With a sufficiently large number of Gaussians in the mixture we can arbitrarily reduce approximation error (Anderson and Moore, 2005); (Alspach and Sorensen, 1972).

The challenge with the Gaussian mixture approximations is to estimate its parameters. We need to find a method to fit a Gaussian mixture to the predictive density of Gaussian undergoing a non-linear transform. Given such a method we can repeat the procedure to obtain a predictive Gaussian mixture for a non-linear transform of the input Gaussian mixture.

The first step to estimate Gaussian mixture parameters to approximate the non-Gaussian, predictive density for a non-linear map of a Gaussian, is described in Section 2.1.1 (Kamthe et al., 2013).

Given the filtering results and all the observations we can improve our estimation result by implementing a backward filter as described in Section 1.2.2. For multi-modal beliefs the optimal backward filter, Equation (1.3), cannot be evaluated in closed form due to conditioning on $p(x_{n+1}|Y_n)$ which is a Gaussian a mixture (see Section 2.1.3). We propose an approximate solution to smoothing with Gaussian mixture beliefs in Section 2.2.

2.1 Multi-Modal Filter

In the following, we devise a closed-form filtering algorithm with multi-modal representations of the state distributions. Our algorithm is inspired by the following observation made by Julier and Uhlmann (Julier and Uhlmann, 2004): “Given only the mean and the variance of the underlying distribution, and, in absence of any a priori information, any distribution (with the same mean and variance) used to calculate the transformed mean and variance is trivially optimal.” This observation was the basis to derive the Unscented transform and the UKF. However, predictions based on the Unscented Transform often under-estimate the true predictive uncertainty, which can result in incoherent state estimation and divergent tracking performance.

To address this issue, we use a different (optimal) representation of the underlying distribution, which still matches the mean and variance: We propose to represent each sigma point in the Unscented transform by a Gaussian centred at this sigma point. This approximation of the original distribution is effectively a GMM with $2D+1$ components, where D is the dimensionality of x.

In Section 2.1.1, we derive an optimal GMM representation of a state distribution $p(x_{n-1})$ of which only the mean and variance are known. In Section 2.1.2, we detail how to map this GMM through a non-linear function to obtain a predictive distribution $p(x_n)$, which is represented by a GMM. We generalise both uncertainty propagation and parameter estimation to the case where $p(x_{n-1})$ is given by a GMM. In Section 2.1.4, we propose a method for pruning the number of mixture components in a GMM to avoid their exponential increase in number of components. In Section 2.1.3, we propose the resulting filtering algorithm, which exploits the results from Sections 2.1.1–2.1.4.

2.1.1 Estimation of the Gaussian Mixture Parameters

Let the mean and the variance of the state distribution $p(x_{n-1})$ be given by μ, Σ, respectively. Then, we can represent $p(x_{n-1})$ by a GMM $p(x_{n-1}) = \sum_{i=0}^{2D} \delta_i \varphi_i(x_{n-1})$, such that the mean and the variance of the approximate density $p(x_{n-1})$ equal the mean μ and variance Σ of $p(x_{n-1})$. This representation is achieved by the closed-form relations

$$\begin{aligned}
&\delta_i = 1/(2D+1), \\
&\mu^0 = \mu, \quad \mu^j = \mu + \sigma^j, \quad \mu^{j+D} = \mu - \sigma^j, \\
&\Sigma^i = \left(1 - \tfrac{2\alpha}{D+1}\right)\Sigma,
\end{aligned} \tag{2.1}$$

where $i = 0, \ldots, 2D$ and $j = 1, \ldots, D$, where D is the dimensionality of the state variable x_{n-1}. The variable σ denotes D rows or columns from the matrix square root $\pm\sqrt{\alpha\Sigma}$. From (2.1), we can see that we need to calculate $\sqrt{\Sigma}$ only once for all $2D+1$ Gaussians $\varphi_i(x_{n-1})$. To ensure that Σ^i is positive semi-definite, the scaling factor α should be chosen such that $2\alpha \leq (2D+1)$, see (2.1). For $2\alpha = 2D+1$ in (2.1), the equations above reduce to scaled sigma points (Julier and Uhlmann, 2004). Hence, the GMM representation in (2.1) can be considered a generalisation of the classical sigma point representation of densities employed by the Unscented Transform, where each sigma point becomes an improper probability distribution.

2.1.2 Propagation of Uncertainty

A key step in filtering is the uncertainty propagation step, i.e. estimating the probability distribution of random variable, which has been transformed by means of the transition function f. Given $p(x_{n-1})$ and the system dynamics (1.4), we determine $p(x_n)$ by evaluating $\int p(x_n|x_{n-1})p(x_{n-1})\mathrm{d}x_{n-1}$. For non-linear functions f, the integral above can not be solved in closed form. Thus, approximate solutions are required.

Uncertainty propagation in non-linear systems can be achieved by approximate methods, employing linearisation or deterministic sampling as in the EKF and UKF. In such approaches, the state distribution $p(x_{n-1})$ and the approximate predictive density $p(x_n)$ are well represented by Gaussians. If the state distribution $p(x_{n-1})$ is a Gaussian mixture as in (2.1), we can estimate the predictive distribution $p(x_n)$ similarly, e.g. by applying such an approximate update to each mixture component in the GMM. In the multi-modal filter, we propagate each mixture component $\varphi_i(x_{n-1})$ of the GMM through f and approximate $p(x_n)$ by

$$\begin{aligned}
p(x_n) &= \int p(x_n|x_{n-1}) \sum_{i=0}^{2D} \delta_i \varphi_i(x_{n-1}) \mathrm{d}x_{n-1} \\
&\approx \sum_{j=0}^{2D} \delta_j \varphi_j(x_n),
\end{aligned} \tag{2.2}$$

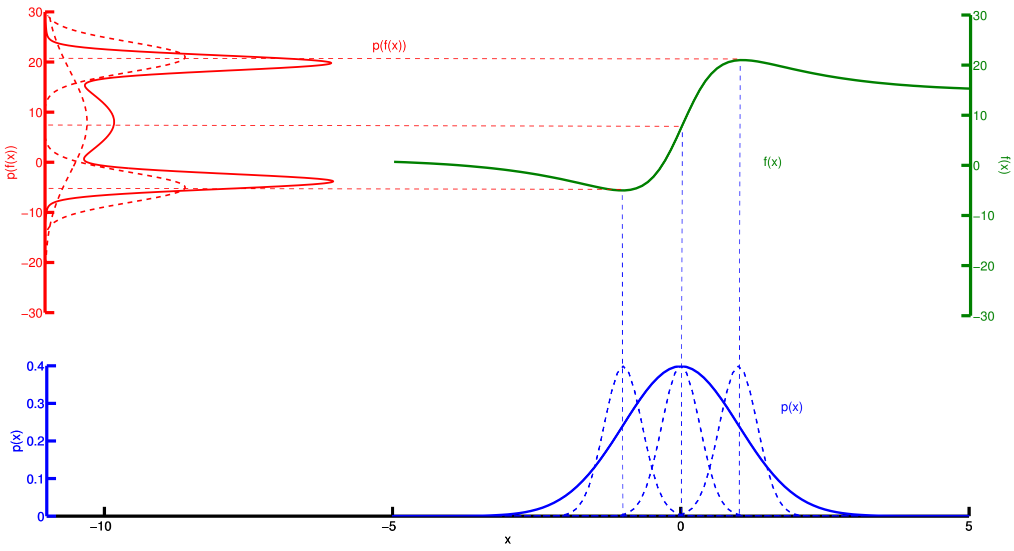


Figure 2.1.: Demonstration of predictive density estimation with multi-modal filter. 1) We approximate Gaussian (blue solid) by a Gaussian mixture (blue dashed) using (2.1) 2) Green solid line represents a non-linear function 3) The solid red line represents a Gaussian mixture evaluated using EM. The dashed red Gaussians represent unscented transform of dashed blue Gaussians. The dashed straight dashed lines represent sigma points used by a unscented transform.

where the mean and covariance of each $\varphi_j(x_n)$ are computed by means of the Unscented Transform. The Figure 2.1, shows an example of uncertainty propagation using scheme described in Equation (2.2).

If the prior density is a Gaussian mixture $p(x_{n-1}) = \sum_{j=0}^{M-1} \beta_j \varphi_j(x_{n-1})$, we repeat the procedure above for each mixture component in $p(x_{n-1})$, i.e. we split each mixture component φ_j into $2D+1$ components $\delta_i \varphi_{ji}$, $i = 0, \dots, 2D$, and propagate them forward using the Unscented Transform. For notational convenience, we define this operation on a Gaussian mixture as $\mathscr{F}_n(f, p(x_{n-1}))$, such that

$$
\begin{aligned}
p(x_n) = \mathscr{F}_n(f, p(x_{n-1})) &= \int p(x_n|x_{n-1}) p(x_{n-1}) \mathrm{d}x_{n-1} \\
&= \sum_{j=0}^{M-1} \sum_{i=0}^{2D} \beta_j \delta_i \int p(x_n|x_{n-1}) \varphi_{ij}(x_{n-1}) \mathrm{d}x_{n-1} \\
&\equiv \sum_{l=0}^{M(2D+1)-1} \gamma_l \varphi_l(x_n),
\end{aligned}
\tag{2.3}
$$

where $\gamma_l = \delta_i \beta_j$. We compute the moments of the mixture components φ_l by means of the Unscented Transform.

From (2.3), it can be observed that during an update step the number of components grows by a factor of M. When applied multiple times, it will result in an exponential explosion of

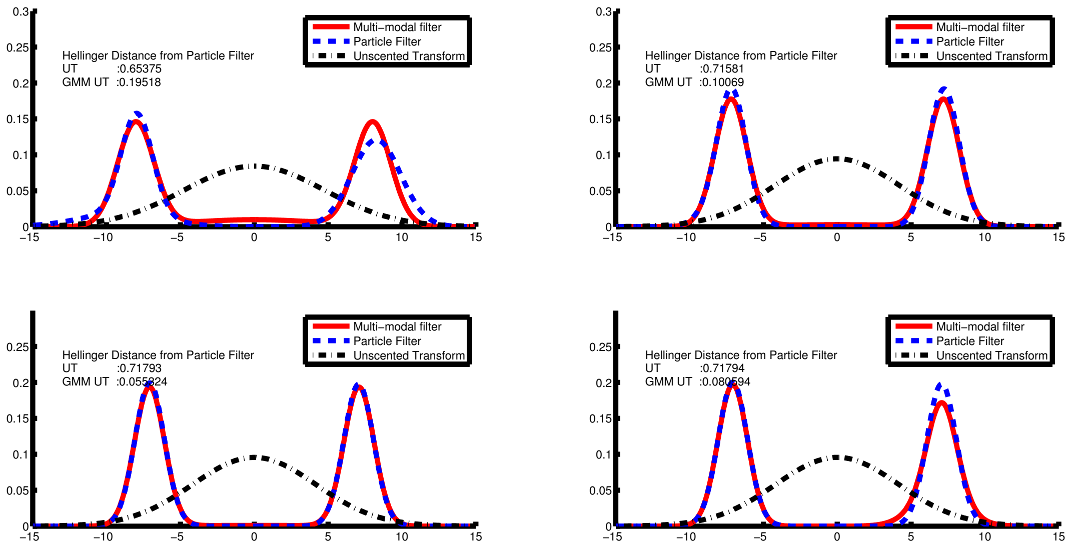


Figure 2.2.: Example of a multi-modal predictive density for a non-linear stationary function with standard normal distribution as prior. The blue curve shows predictive distribution with calculated using EM (10^5 samples, 3 components). The solid red curve shows fit with proposed mulit-modal method and black solid line represents the Unscented Transform approximation of predictive density. We use Hellinger distance(Beran, 1977) as a metric to measure goodness of fit which shows that proposed method outperforms standard Unscented Transform based method for this non-linear function. We can also visually confirm that multi-modal filter is able to capture both modes of the distribution.

components. We reduce the number of mixture components at each time step, for details, see Section 2.1.4.

2.1.3 Filtering

In the following, we subsume all derivations in our multi-modal non-linear state estimator, whose time and measurement updates are summarised in the following.

Time Update

Assume that the filter distribution $p(x_{n-1}|Y_{n-1})$ is represented by a GMM with M components. The time update, i.e. the one-step ahead predictive distribution is given by

$$p(x_n|Y_{n-1}) = \int p(x_n|x_{n-1})p(x_{n-1}|Y_{n-1})\,\mathrm{d}x_{n-1}. \tag{2.4}$$

This integral can be evaluated as $\mathscr{F}_n(f, p(x_{n-1}|Y_{n-1}))$, such that we obtain a GMM representation of the time update

$$p(x_n|Y_{n-1}) = \sum_{j=0}^{M(2D+1)-1} \gamma_j \varphi_j(x_{n|n-1}) \tag{2.5}$$

as detailed in (2.3).

Measurement Update

The measurement update can be approximated up to a normalisation constant by

$$p(x_n|Y_n) \propto p(y_n|x_n)p(x_n|Y_{n-1}), \tag{2.6}$$

where $p(x_n|Y_{n-1})$ is the time update (2.5). We now apply a similar operation as in (2.3) with the measurement function h as a non-linear function and obtain

$$p(y_n|Y_{n-1}) = \mathscr{F}_n(h, p(x_n|Y_{n-1})). \tag{2.7}$$

Substituting (2.7) and (2.5) in (2.6) yields the measurement update, i.e. the filtered state distribution

$$\begin{aligned} p(x_{n|n}) &\propto \sum_{i=0}^{2D} \delta_i \varphi_i(y_{n|n-1}) \sum_{j=0}^{M(2D+1)-1} \gamma_j \varphi_j(x_{n|n-1}), \\ &\equiv \sum_{l=0}^{M(2D+1)^2-1} \beta_l \varphi_l(x_{n|n}). \end{aligned} \tag{2.8}$$

We calculate the measurement update for each pair φ_i and φ_j. Recalling that $\varphi_l(x_{n|n}) = \mathscr{N}(x|\mu_{n|n}^{ij}, \Sigma_{n|n}^{ij})$ for $i = 0, \ldots, 2D$ and $j = 0, \ldots, (2D+1)^2 - 1$, the measurement updates (Särkkä, 2008) and weight updates (Gaussian Sum (Alspach and Sorenson, 1972)) are given by

$$\begin{aligned} K_n^j &= \Gamma_{n|n-1}^j \left(\Sigma_{n|n-1}^j\right)^{-1}, \\ \mu_{n|n}^{ij} &= \mu_{n|n-1}^i + K_n^j(y - \mu_{n|n-1}^j), \\ \Sigma_{n|n}^{ij} &= \Sigma_{n|n-1}^i - K_n^j(\Sigma_{n|n-1}^j) {K_n^j}^T, \\ \beta_{i,j} &= \frac{\delta_i \gamma_j \mathscr{N}(x = y \mid \mu_{n|n-1}^j, \Sigma_{n|n-1}^j)}{\sum_{k,l} \delta_l \gamma_k \mathscr{N}(x = y \mid \mu_{n|n-1}^k, \Sigma_{n|n-1}^k)}, \end{aligned} \tag{2.9}$$

where $\Gamma_{n|n-1}^j$ is the cross covariance matrix $\text{cov}(x_{n-1}, x_n)$ determined via the Unscented Transform (Särkkä, 2008).

After the measurement update, we reduce the $M(2D+1)^2$ mixture components in the GMM, see (2.8), to M according to Section 2.1.4.

2.1.4 Mixture Reduction

Up to this point, we have considered the case where a density with known mean and variance has been represented by a GMM, which could subsequently be used to estimate the predicted state distribution. Incorporating these steps into an recursive state estimator for time series, there is an exponential growth in the number of mixture components in (2.3). One way to mitigate this effect is to represent the estimated densities by a mixture model with a fixed number of components (Anderson and Moore, 2005). To keep the number of mixture components constant we can reduce them at each time step (Anderson and Moore, 2005).

A straightforward and fast approach is to drop the Gaussian components with the lowest weights. Such omissions, however, can result in poor performance of the filter (Kitagawa, 1994). Kitagawa (Kitagawa, 1994) suggested to repeatedly merge a pair Gaussian components. A pair is selected with lowest distance in terms of some distance metric. We evaluated multiple distance metrics, e.g. the L_2 distance (Williams and Maybeck, 2003), the KL divergence (Runnalls, 2007), and the Cauchy Schwarz divergence (Kampa et al., 2011). In this report, we used the symmetric KL divergence (Kitagawa, 1994), $D(p,q) = (KL(p|q) + KL(q|p))/2$, which outperformed aforementioned distance measures for mixture reduction in filtering.

2.2 Multi-Modal Smoothing

The smoothing update is based on the forward-backward filter. The optimal backward pass for a dynamical system described by Figure 1.3 is given by equation (1.3) (Kitagawa, 1994), we repeat the equation (1.3) here for quick reference,

$$p(x_n|Y_N) = p(x_n|Y_n) \int \frac{p(x_{n+1}|Y_N)p(x_{n+1}|x_n)}{p(x_{n+1}|Y_n)} \mathrm{d}x_{n+1}. \tag{2.10}$$

Equation (2.10) is a recursion initiated by setting $p(x_{n+1}|Y_N) = p(x_{n+1}|Y_{n+1})$ where the index $n+1$ now points at last observation, i.e. $N = n+1$. The quantities we know are

$$p(x_{n+1}|Y_N) = \sum_j^M \alpha_j \varphi_j(x_{n+1}|Y_N), \qquad \text{previous smoothing result} \tag{2.11}$$

$$p(x_n|Y_n) = \sum_i^P \delta_i \varphi_i(x_n|Y_n), \qquad \text{previous filtering result} \tag{2.12}$$

$$p(x_{n+1}|x_n) = \sum_k^L \alpha_k \varphi_k(x_{n+1}|x_n). \qquad \text{forward prediction or time update} \tag{2.13}$$

The denominator of (2.10) is a Gaussian mixture that can be evaluated as am integral $\int p(x_{n+1}|x_n)p(x_n|Y_n)\mathrm{d}x_n$. A Gaussian mixture in the denominator makes a closed form exact solution to the equation (2.10) impossible and we need approximations for a Gaussian sum smoother.

We propose an approximate solution to the Gaussian sum smoother in Section 2.2.1. The solution proposed in Section 2.2.1 is not available in the literature to the best of our knowledge. In the following, we list state of the art deterministic approximations to the smoothing Equation (2.10).

- Kitagawa's two filter smoother (Kitagawa, 1994). In the two filter smoother the right hand side integral is replaced by a backward information filter $p(Y_N|x_n)$, i.e. we write the smoothing pass as, $p(x_n|Y_N) \propto \ p(x_n|Y_n)\ p(Y_N|x_n)$.

- Closed form forward-backward smoothing (Vo et al., 2012). The forward-backward smoother uses a recursion to calculate likelihood to approximate smoother as, $p(x_n|Y_N) = p(x_n|Y_n)\ Lik(Y_N|x_n)$, the $Lik(\ )$ denotes likelihood function. The forward-backward replaces the information filter in Kitagawa's two filter smoother by the likelihood and avoids the inversion of transformation matrix (Vo et al., 2012)

- Expectation Correction (Barber, 2006). The expectations correction algorithm ignores the integral altogether and evaluates left hand side at the mean values of Gaussians in mixture, i.e. we replace Gaussian by its mean. The crude approximation gives excellent results, underlining the importance of a Gaussian sum smoother in estimation.

The Gaussian sum smoothers described above attempt to solve the problem of backward pass with Gaussian mixture beliefs, however, the system is assumed to be linear (Switching Linear Dynamical Systems (SLDS)). There is no smoothing algorithm in literature for inference in a non-linear dynamical system with multi-modal beliefs, to the best of our knowledge.

The proposed multi-modal smoother is, however, not restricted to non-linear dynamical systems alone and in principle can be applied to SLDS as well.

2.2.1 Gaussian Sum Smoother for Non-linear systems

Proposition 2.1. *If we define probabilities as in Equations* (2.11) *to* (2.13)*, the Gaussian mixture approximation to smoothing equation* (2.10) *is given as*

$$p(x_n|Y_N) = \sum_{z=1}^{LMP} \beta_z \mathcal{N}(x_{n|N}|\mu^z_{n|N}, \Sigma^z_{n|N}), \tag{2.14}$$

where the mean $\mu^z_{n|N}$ and covariance $\Sigma^z_{n|N}$ are given by

$$\begin{aligned}
D^k_n &= \Gamma^k_{n+1|n}\left(\Sigma^k_{n+1|n}\right)^{-1}, \\
\mu^{ijk}_{n|N} &= \mu^i_{n|n} + D^k_n\left(\mu^j_{n+1|N} - \mu^k_{n+1|n}\right), \\
\Sigma^{ijk}_{n|N} &= \Sigma^i_{n|n} + D^k_n\left(\Sigma^j_{n+1|N} - \Sigma^k_{n+1|n}\right) D^{k^T}_n, \\
\beta_{i,j,k} &= \frac{\delta_i\,\alpha_j\,\gamma_k\,\mathcal{N}(x = \mu^j_{n+1|N}|\,\mu^k_{n|n-1}, \Sigma^k_{n+1|n} + \Sigma^j_{n+1|N})}{\sum_{p,q,r}\delta_p\,\alpha_q\,\gamma_r\,\mathcal{N}(x = \mu^q_{n+1|N}|\,\mu^r_{n+1|n}, \Sigma^r_{n+1|n} + \Sigma^q_{n+1|N})}.
\end{aligned} \tag{2.15}$$

Proof. We can rewrite (2.10) as

$$p(x_n|Y_N) = \int \underbrace{\underbrace{p(x_n|x_{n+1}, Y_N)}_{\text{Step 1}} p(x_{n+1}|Y_N)\ \mathrm{d}x_{n+1}}_{\text{Step 2}}. \tag{2.16}$$

Step 1

The conditioning is defined as

$$p(x_n|x_{n+1}, Y_N) = \frac{p(x_{n+1}|x_n)p(x_n|Y_n)}{p(x_{n+1}|Y_n)}. \tag{2.17}$$

The right hand side of (2.17) cannot be evaluated in closed form as the denominator $p(x_{n+1}|Y_n)$ is a Gaussian mixture given by equation (2.13). We rewrite the denominator as,

$$p(x_n|x_{n+1}, Y_N) = \frac{p(x_{n+1}|x_n)p(x_n|Y_n)}{\int p(x_{n+1}|x_n)p(x_n|Y_n)\ \mathrm{d}x_n}. \tag{2.18}$$

The expression $\int p(x_{n+1}|x_n)p(x_n|Y_n)\ \mathrm{d}x_n$ in the denominator is independent of x_n and can be treated as normalisation constant for the probability distribution $p(x_n|x_{n+1}, Y_N)$. We can obtain

$$p(x_n|x_{n+1}, Y_N) \propto\ p(x_{n+1}|x_n)\ p(x_n|Y_n) = \beta\ p(x_{n+1}|x_n)\ p(x_n|Y_n) \tag{2.19}$$

where β is a constant independent of x_n and defined as

$$\beta\ = 1 \bigg/ \int p(x_{n+1}|x_n)p(x_n|Y_n)\ \mathrm{d}x_n. \tag{2.20}$$

It should be noted that β is a function of x_{n+1} and, hence, cannot be treated as a constant when marginalising over x_{n+1}. Substituting (2.13) in (2.19) yields

$$p(x_n|x_{n+1}, Y_N) = \beta\ \sum_k^L \gamma_k \varphi_k(x_{n+1}|x_n)\ \sum_j^M \alpha_j \varphi_j(x_{n+1}|Y_n). \tag{2.21}$$

This derivation completes Step 1 of (2.16). Substituting the result of Step 1 (2.17) in (2.16) we obtain

$$p(x_n|Y_N) = \int \beta\ \sum_k^L \gamma_k \varphi_k(x_{n+1}|x_n)\ \sum_i^P \delta_i \varphi_i(x_n|Y_n) \sum_j^M \alpha_j \varphi_j(x_{n+1}|Y_N)\ \mathrm{d}x_{n+1}. \tag{2.22}$$

Pushing the Gaussian variables φ_k and φ_i inside the sum yields

$$p(x_n|Y_N) = \int \beta \sum_k^L \gamma_k \sum_i^P \delta_i \sum_j^M \alpha_j \underbrace{\underbrace{\varphi_k(x_{n+1}|x_n)\ \varphi_i(x_n|Y_n)}_{\varphi_{i,k}(x_n, x_{n+1}|Y_n)} \varphi_j(x_{n+1}|Y_N)}_{\text{Step 2}}\ \mathrm{d}x_{n+1}. \tag{2.23}$$

We define the individual components of Gaussian mixture φ as

$$\varphi_i(x_n|Y_n) = \mathcal{N}(x_{n|n}|\mu^i_{n|n}, \Sigma^i_{n|n}), \tag{2.24}$$

$$\varphi_k(x_{n+1}|Y_n) = \mathcal{N}(x_{n+1|n}|\mu^k_{n+1|n}, \Sigma^k_{n+1|n}). \tag{2.25}$$

to evaluate the joint distribution

$$\varphi_{i,k}(x_n, x_{n+1}|Y_n) = \mathcal{N}(x'_{n|n}|\mu^{i,k}_{n|n}, \Sigma^{i,k}_{n|n}), \tag{2.26}$$

where

$$\begin{aligned} D^k_n &= \Gamma^k_{n+1|n}\left(\Sigma^k_{n+1|n}\right)^{-1}, \\ \mu^{i,k}_{n|n} &= \mu^i_{n|n} + D_n(x_{n+1|n} - \mu^k_{n+1|n}), \\ \Sigma^{i,k}_{n|n} &= \Sigma^i_{n|n} + D_n(\Sigma^k_{n+1|n})D^T_n. \end{aligned} \tag{2.27}$$

The covariance Σ and cross-covariance Γ can be obtained by any method non-linear Gaussian approximation methods like Unscented Transform, extended Kalman filter etc.
By definition of β in (2.20) we write Gaussian component of conditional $p(x_n, x_{n+1}|Y_n)$ as,

$$\varphi_{i,k}(x_n|x_{n+1}, Y_n) = \varphi_{i,k}(x_n, x_{n+1}|Y_n)\ \beta(x_{n+1}) \tag{2.28}$$

Substituting values from(2.28) in (2.23) yields

$$p(x_n|Y_N) = \int \beta(x_{n+1}) \sum_k^L \gamma_k \sum_i^P \delta_i \sum_j^M \alpha_j \underbrace{\underbrace{\varphi_{i,k}(x_n|x_{n+1}, Y_n)}_{\text{Step 1}} \varphi_j(x_{n+1}|Y_N)}_{\text{Step 2}}\ \mathrm{d}x_{n+1}. \tag{2.29}$$

Step 2 We marginalise over x_{n+1} to obtain

$$\varphi_{i,j,k}(x_n|x_{n+1}, Y_n) = \mathcal{N}(x_{n|n}|\mu^{i,j,k}_{n|n}, \Sigma^{i,j,k}_{n|n}), \tag{2.30}$$

where

$$D_n^k = \Gamma_{n+1|n}^k \left(\Sigma_{n+1|n}^k\right)^{-1}, \tag{2.31}$$
$$\mu_{n|n}^{i,j,k} = \mu_{n|n}^i + D_n(\mu_{n+1|N}^j - \mu_{n+1|n}^k),$$
$$\Sigma_{n|n}^{i,j,k} = \Sigma_{n|n}^i + D_n\Sigma_{n+1|N}^j - \Sigma_{n+1|n}^k D_n^T.$$

To evaluate the constant $\beta(x_{n+1})$, we use Theorem 1 to write

$$\beta(x_{n+1}) = \mathcal{N}(x_{n+1}|\mu_{n+1|n}^k, \Sigma_{n+1|n}^k) \tag{2.32}$$

and using (2.29) we obtain

$$\beta_{i,j,k} = \delta_i\ \alpha_j\ \gamma_k \int\ \mathcal{N}(x_{n+1}|\mu_{n+1|n}^k), \Sigma_{n+1|n}^k)\ \mathcal{N}(x_{n+1|n}|\mu_{n+1|N}^j, \Sigma_{n+1|N}^j)\ \mathrm{d}x_{n+1}. \tag{2.33}$$

We use definition of product of two Gaussians, and the fact that marginalisation is over x_{n+1} we obtain

$$\begin{aligned}\beta_{i,j,k} &= \delta_i\ \alpha_j\ \gamma_k\ \mathcal{N}(x = \mu_{n+1|N}^j|\mu_{n+1|n}^k, \Sigma_{n+1|n}^k + \Sigma_{n+1|N}^j) \int \mathcal{N}(x_{n+1}|, \mu_{n+1}, \Sigma_{n+1})\ \mathrm{d}x_{n+1}, \quad (2.34)\\ &= \delta_i\ \alpha_j\ \gamma_k\ \mathcal{N}(x = \mu_{n+1|N}^j|\mu_{n+1|n}^k, \Sigma_{n+1|n}^k + \Sigma_{n+1|N}^j).\end{aligned}$$

To ensure the Gaussian mixture approximation is a valid probability distribution we must ensure that all weights sum to one. Therefore, we normalise β according to

$$\beta_{i,j,k} = \frac{\delta_i\,\alpha_j\,\gamma_k\,\mathcal{N}(x = \mu_{n+1|N}^j|\ \mu_{n|n-1}^k, \Sigma_{n+1|n}^k + \Sigma_{n+1|N}^j)}{\sum_{p,q,r} \delta_p\,\alpha_q\,\gamma_r\,\mathcal{N}(x = \mu_{n+1|N}^q|\ \mu_{n+1|n}^r, \Sigma_{n+1|n}^r + \Sigma_{n+1|N}^q)}. \tag{2.35}$$

The right hand side of (2.29) is completed by (2.35) and (2.31) and completes the proof of Proposition 2.1.

□

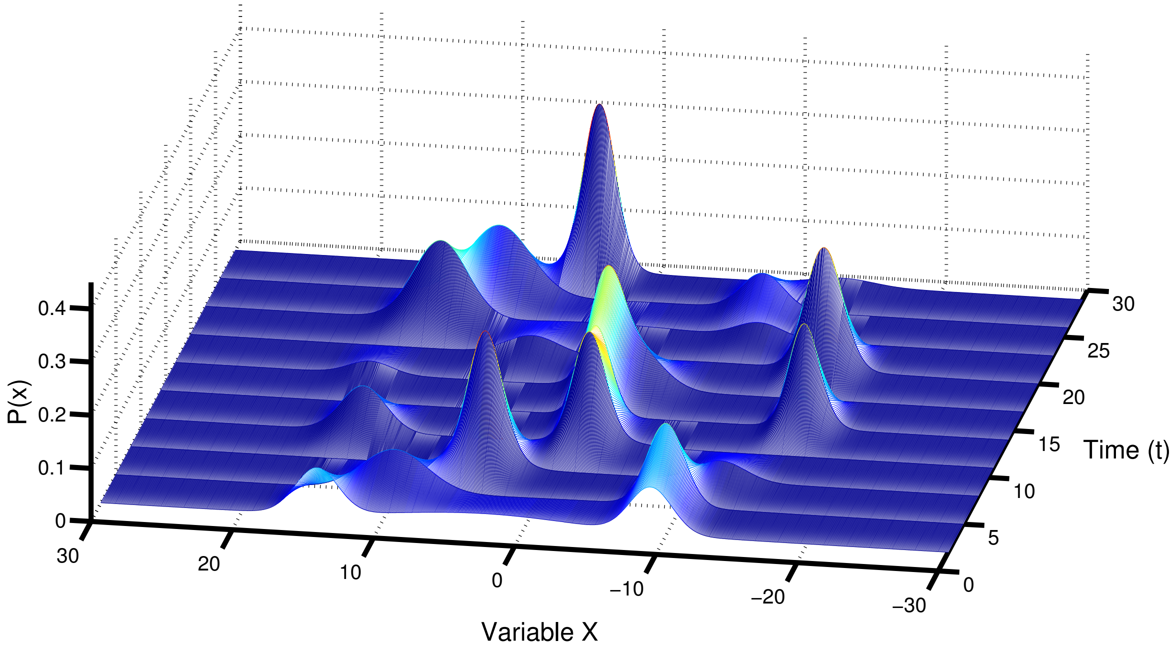
2.3 Summary of the Multi-modal filter and Smoother

In this chapter we proposed a multi-modal filter and multi-modal smoother based on the Gaussian sum approximation. The important points of this chapter are below.

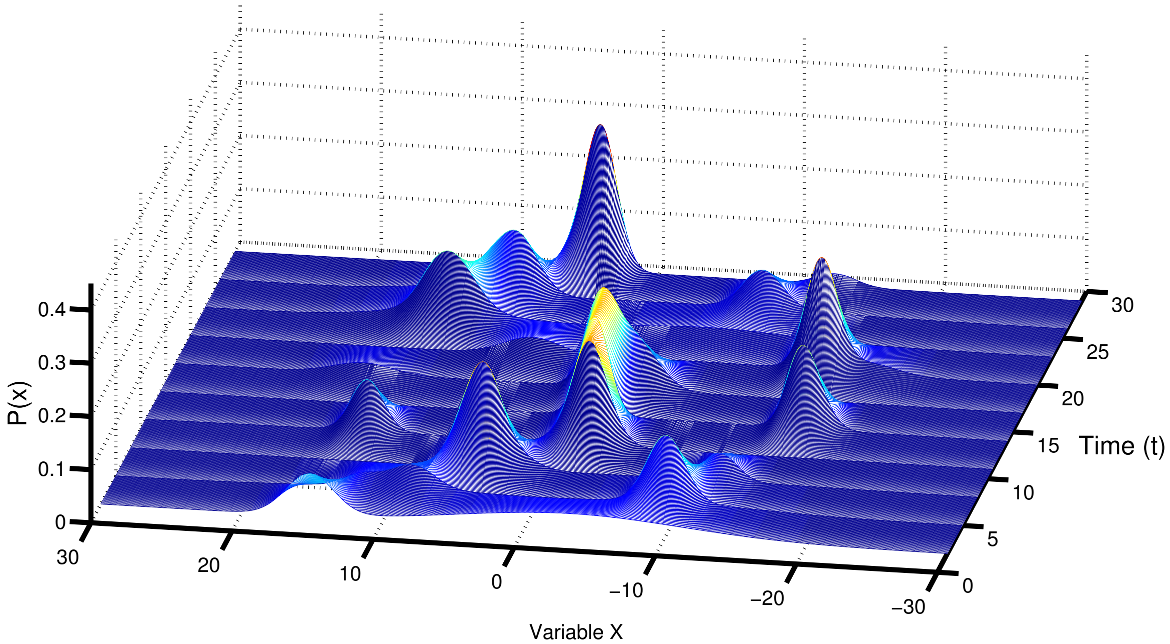
- We described a closed form solution for estimating the mixture parameters of the density obtained, when mapping a Gaussian mixture through a non-linear mapping. The approximations of a Gaussian by a Gaussian mixture is based on Cholesky factorisation and the approximation can be described as a generalisation of the Unscented Transform. The relative error values in likelihood per data point of Gaussian samples, under Gaussian mixture as a model shows that our approximation does not introduce large errors (see figure).

- We used Gaussian sum filter for filtering with multi-modal beliefs, the important property of Gaussian sum filter is that it is based on the *Gaussian Sum Approximations*. A Gaussian sum approximation allows a filter to match the optimal Bayesian filter as the number of mixture components increases (Anderson and Moore, 2005). Hence, our filter based on Gaussian sum filter will converge to the optimal Bayesian filter as the number of components reaches infinity.
- We use the Gaussian sum approximations to devise a closed form Gaussian sum smoother.
- Mixture reduction: We use the symmetric KL divergence as a measure for a distance based Gaussian mixture reduction technique. It was found to perform better than standard mixture reduction techniques. A comprehensive study on the various mixture reduction techniques is marked as a future work.

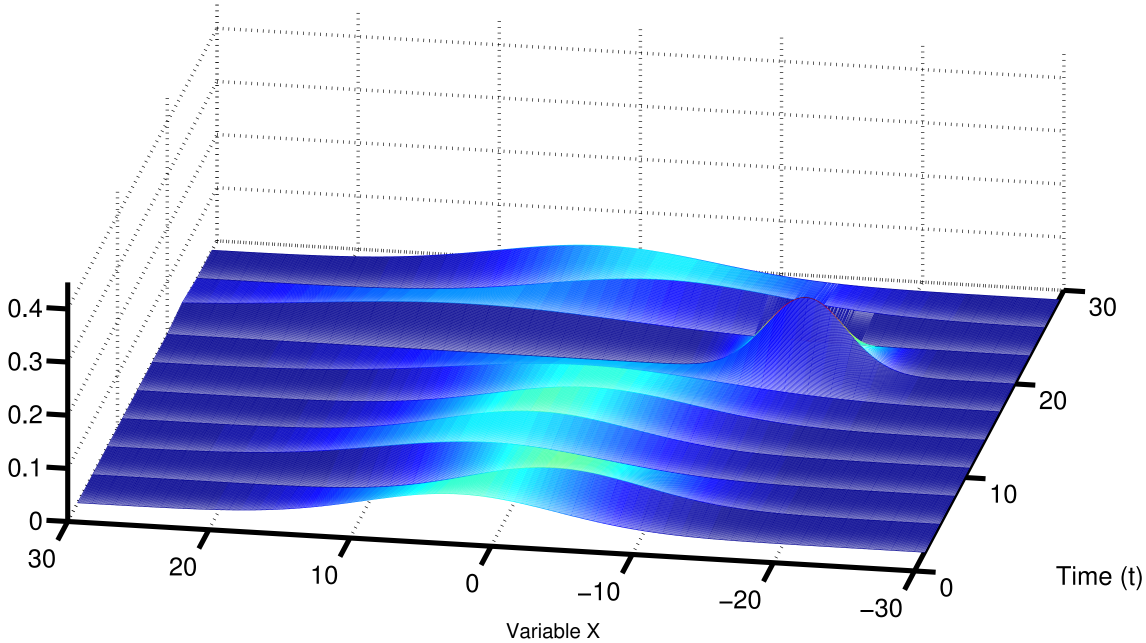
The multi-modal filter and smoother allow us to use the flexibility of Gaussian mixture models over Gaussian models. The parametric form makes filter deterministic and, hence, the numerical results we obtain in next next chapter are reproducible.



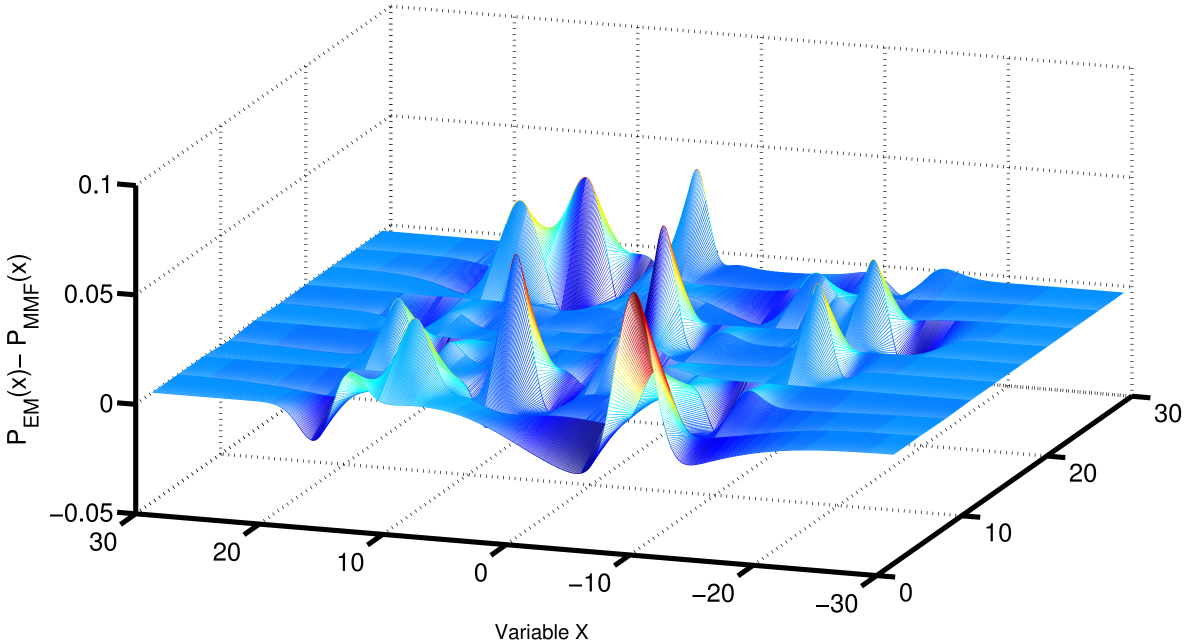
a) We generate 10^6 samples from prior $p(x_{n-1})$ propagate density forward by using $x_n = f(x_{n-1})$ to obtain 10^6 samples of $p(x_n)$ and use Expectation Maximisation (EM) to approximate $p(x_n)$ with M=3 components



(b) We calculate density propagation $p(x_n|x_{n-1})$ using the proposed Multi-Modal filter with $\alpha = 1.3$ and the density $p(x_n) \sim p(x_n|x_{n-1})$ is represented by using M=3 components



(c) Unscented Transform approximation of $p(x_n) \sim p(x_n|x_{n-1})$



(d) Absolute error between plots a) & b)

Figure 2.3.: The Figure demonstrates the ability of the proposed multi-modal approximations for non-stationary non-linear function. The plots a) and b) show EM and Multi-modal approximations, to facilitate comparison both plots use same Y axis and colour coding. Visual inspection shows that the proposed filter is able to track predictive densities, we confirm this by plotting the error between EM estimate and multi-modal estimate. The Y axis is different for plot d).

3 Numerical Results

We evaluated our proposed filtering algorithm on data generated from standard one-dimensional non-linear dynamical systems. Both the UKF and the Multi-Modal Filter (M-MF) use the same parameters for the Unscented Transform, i.e. $\alpha = 1$, $\beta = 2$ and $\kappa = 2$. The prior $p(x_0)$ is a standard Gaussian $p(x_0) = \mathcal{N}(x|0,1)$. Densities in the M-MF are represented by a Gaussian mixture with $M = 3$ components. The mean of the filtered state is estimated by the first moment of this Gaussian mixture. The employed Particle filter (PF) is a standard particle filter with residual re-sampling scheme. The PF-UKF (Van Der Merwe et al., 2000) on the other hand uses the Unscented Transform as proposal distribution and the UKF for filtering. Unlike our M-MF the PF-UKF uses random sampling and the UKF. We use the root mean square error (RMSE) and the predictive Negative Log-Likelihood (NLL) per observation as metrics to compare the performance of the different filters. Lower values indicate better performance. The results in Table 3.1 were obtained from 100 independent simulations with $T = 100$ time steps for each of the following system models.

3.1 Univariate Non-linear Growth Model

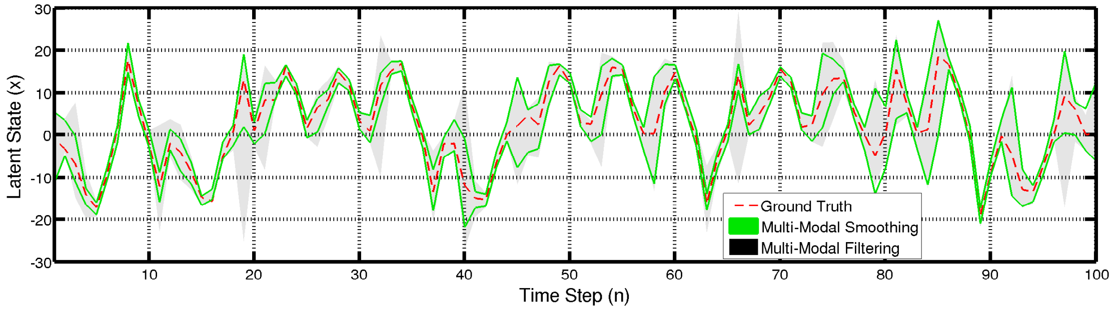
The the Univariate Non-stationary Growth Model (UNGM) (Doucet et al., 2001) is a standard benchmark problem for non-linear estimators.

3.1.1 Non-stationary Model

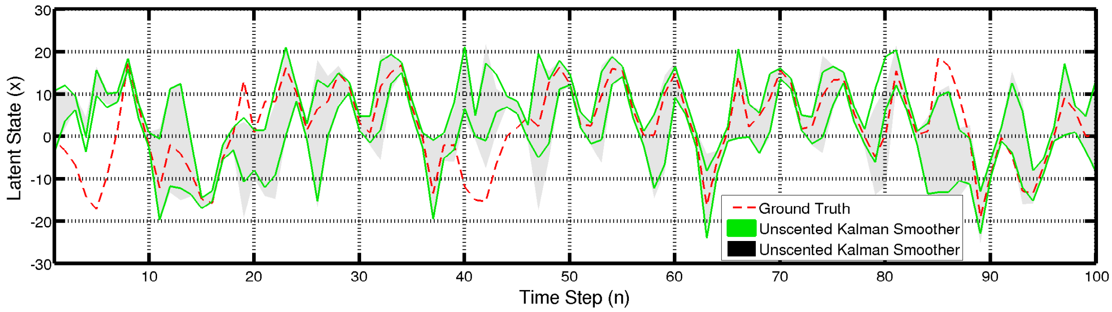
We tested the different filters on a standard system, the Univariate Non-stationary Growth Model (UNGM) from (Doucet et al., 2001)

$$\begin{aligned} x_n &= \frac{x_{n-1}}{2} + \frac{25x_{n-1}}{1+x_{n-1}^2} + 8\cos(1.2(n-1)) + w, & w &\sim \mathcal{N}(0,1), \\ y_n &= \frac{x_n^2}{20} + v, & v &\sim \mathcal{N}(0,1). \end{aligned} \tag{3.1}$$

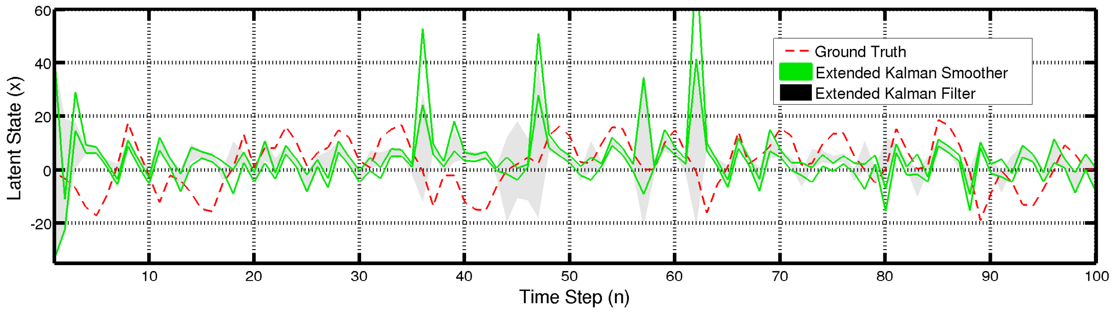
The true state density of the non-stationary model stated above alternated between multi-modal and uni-modal distributions. The switch from uni-modal to a bi-modal density occurred when the mean was close to zero. The quadratic measurement function makes it difficult to distinguish between the two modes as they are symmetric around zero. This symmetry posed a substantial challenge for several filtering algorithms. The PF lost track of the state as $N = 500$ particles failed to capture the true density especially in its tails, which led to degeneracy (Doucet et al., 2001). The particle filters in Table 3.1 are in their standard form and performance may improve if advanced techniques are used (Doucet et al., 2001), as can be seen from the Gibbs filter (Deisenroth and Ohlsson, 2011). The proposed M-MF could track both modes and, hence, led to more consistent estimates. The RMSE performance of the multi-modal filter (M-MF)



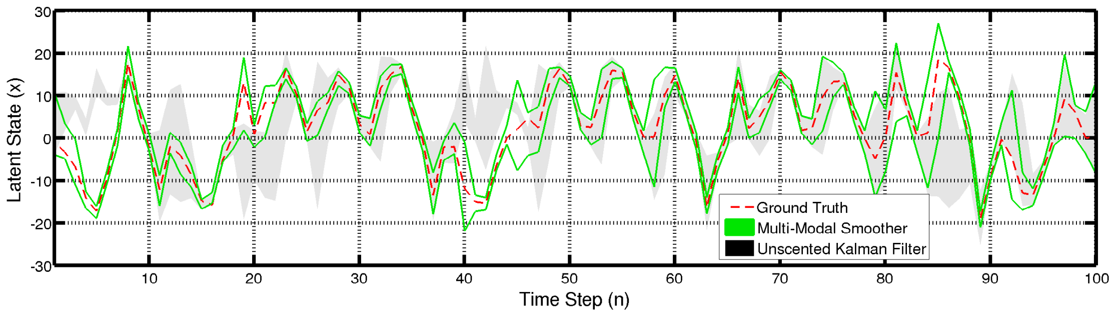
(a) Multi-modal filter (Multi-modal smoother). RMSE **2.58 (1.28)**. NLL **1.79 (1.57)**



(b) Unscented Kalman filter (Unscented Kalman smoother). RMSE 7.738 (7.734). NLL 13.12 (13.44)



(c) Extended Kalman Filter (Extended Kalman smoother). RMSE 10.07 (13.42). NLL 90.82 (118.34)



(d) Unscented Kalman filter (Multi-modal Smoother). RMSE 7.738 **(1.33)**. NLL 13.12 **(1.577)**

Figure 3.1.: An example trajectory from table 3.1 for non-stationary quadratic measurement function (UNGM). In Figure d) the forward pass is UKF and the multi-modal smoother is used for backward pass. The true state is represented by dashed red line, the shaded region represents 95% confidence area for filtering distribution and solid green lines represent 95% confidence area for smoothing distribution

	Stationary $h(x)=5\sin(x)$		Non-Stationary $h(x)=x^2/20$		Non-Stationary $h(x)=5\sin(x)$	
	RMSE	NLL	RMSE	NLL	RMSE	NLL
EKF	7.59 ± 0.31	329.22 ± 27.13	11.03 ± 0.88	101.7 ± 21.3	10.31 ± 0.31	630.9 ± 60.23
UKF	12.27 ± 2.3	89.95 ± 44.96	6.77 ± 1.22	14.9 ± 7.69	9.96 ± 1.69	37.48 ± 20.6
M-MF	$\mathbf{0.97 \pm 0.17}$	$\mathbf{1.375 \pm 0.1595}$	$\mathbf{3.48 \pm 0.56}$	$\mathbf{2.03 \pm 0.7}$	$\mathbf{6.4 \pm 1.73}$	$8.65.7 \pm 8.04$
PF	2.4 ± 0.03	N/A	3.5 ± 0.86	17.4 ± 8.3	11.2 ± 3.6	N/A
Gibbs Filter	3.62 ± 0.4	3.06 ± 0.03	3.97 ± 0.4	2.18 ± 0.1	8.64 ± 0.4	$\mathbf{3.57 \pm 0.08}$

Table 3.1.: Average performances of the filters are shown along with standard deviation. Lower values are better. The M-MF filter performs better than the standard filtering methods.

	Stationary $h(x)=5\sin(x)$		Non-Stationary $h(x)=x^2/20$		Non-Stationary $h(x)=5\sin(x)$	
	RMSE	NLL	RMSE	NLL	RMSE	NLL
EKS	10.16 ± 0.27	$5.42 \times 10^3 \pm 1.55 \times 10^3$	12.2 ± 1.19	210.8 ± 81.9	10.4 ± 0.4	662.2 ± 98.5
UKS	12.40 ± 2.66	91.72 ± 45.73	6.42 ± 1.24	15.9 ± 8.13	10.0 ± 24	44.79 ± 25.6
M-MS	$\mathbf{0.92 \pm 0.13}$	$\mathbf{1.378 \pm 0.1593}$	$\mathbf{1.91 \pm 0.59}$	$\mathbf{1.67 \pm 0.6}$	$\mathbf{6.5 \pm 1.43}$	$\mathbf{10.92 \pm 9.06}$

Table 3.2.: Average performances of the smoothers are shown along with standard deviation. Lower values are better. The M-MS filter performs better than the standard smoothing methods.

is significantly better than the UKF, with the M-MF outperforming the UKF in terms of a lower mean error and standard deviation. The main advantage of the M-MF is its ability to capture the uncertainty appropriately. The NLL values of the M-MF were significantly better than the UKF even when the same parameters are used to calculate the Unscented Transform, see Table 3.1.

We tested the UNGM with an alternative measurement function $h(x) = 5\sin(x)$. For this function, the performance of the EKF is best in terms of RMSE, since its estimates are more stable. The proposed M-MF could track multiple modes, which resulted in significantly better performance in terms of NLL. Moreover, the filter performance was consistently stable, indicated by the small standard deviation values for the NLL measure.

3.1.2 Stationary Model

The stationary model is based on the Uniform Non-stationary Growth Model (UNGM) described above by (Kitagawa, 1996) and can be described by the following equations:

$$f(x) = \frac{x}{2} + \frac{25x}{1+x^2} + w, \qquad w \sim \mathcal{N}(0,1), \tag{3.2}$$

$$h(x) = 5\sin(x) + v, \qquad v \sim \mathcal{N}(0,1). \tag{3.3}$$

We see from the results in Table 3.1 that the UKF was outperformed by all other deterministic filters. The failures of the UKF and the PF-UKF are attributed to their overconfident predictions for sinusoidal functions, which confirms the results in (Deisenroth et al., 2012). The EKF approximates these sinusoidal functions better but both the UKF and EKF fail to capture the multi-modal nature of system dynamics. Thus, the EKF and UKF are inconsistent for the model and settings used in this experiment. The proposed M-MF on the other hand performed consistently better in terms of RMSE and NLL values (see table 3.1). Moreover, the small standard deviation of the NLL suggests that our proposed M-MF is consistent and stable.

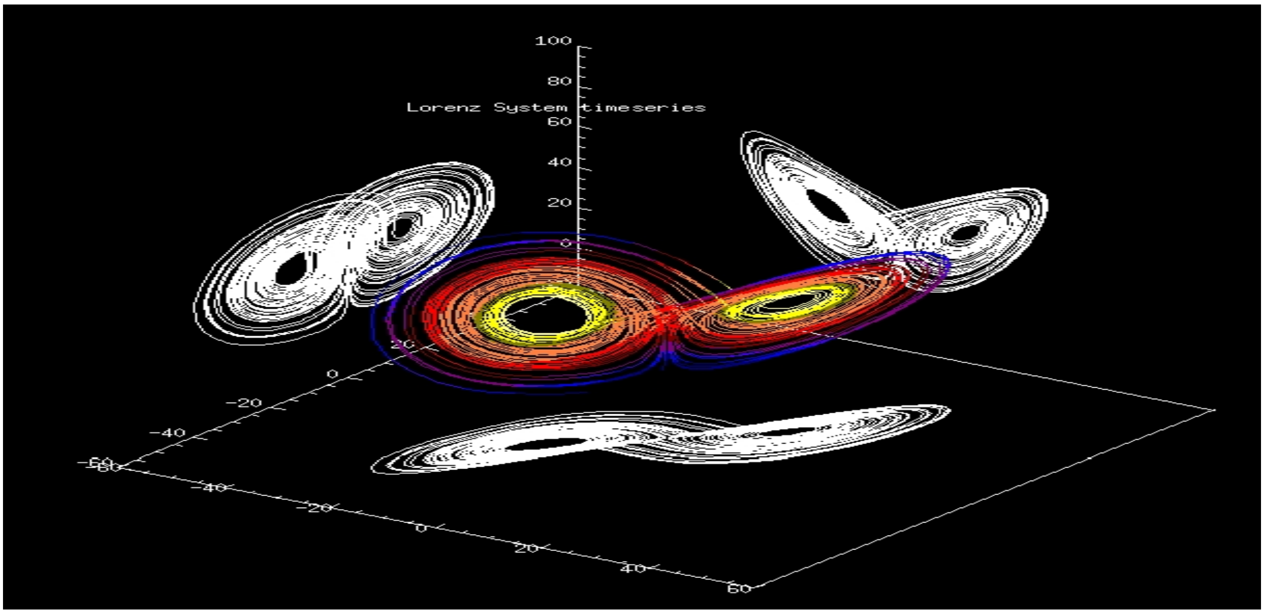


Figure 3.2.: Lorenz time series (reproduced from `http://math.gmu.edu/~rsachs/tj/`)

3.2 Lorenz System

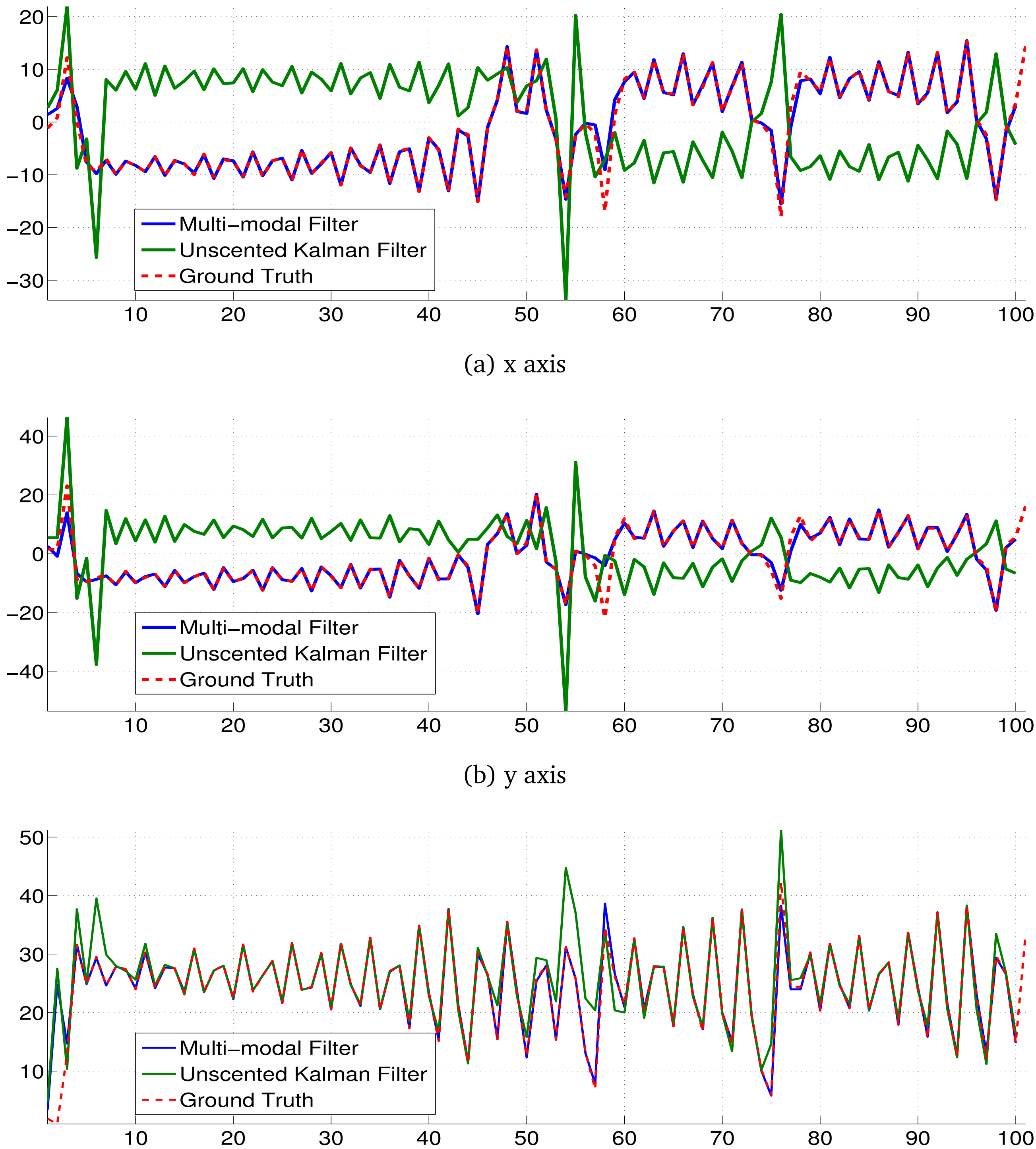
The Lorenz system is a non-linear dynamical system proposed by Lorenz in (Lorenz, 1963) as a solution to the heat convection flow in atmosphere. This system is a 3 dimensional, non-linear, deterministic dynamical system defined by coupled deferential equations (3.4) (Lorenz, 1963) (Hilborn, 2000). The system is unstable for certain values of parameters and can be described as deterministic chaos (Hilborn, 2000). The chaotic behaviour is characterised by an unstable system highly sensitive to initial values, and a small deviation may result in a completely different evolution of system states (Lorenz, 1963). We use the parameter values $\rho = 28$, $\sigma = 10$, and $\beta = 8/3$ as described by Lorenz in (Lorenz, 1963). The system is bistable and describes a butterfly shape usually associated with non-linear chaos (Hilborn, 2000). The system is described by the differential equations

$$
\begin{aligned}
\frac{\partial x}{\partial t} &= \sigma(y - x), \qquad (3.4)\\
\frac{\partial y}{\partial t} &= x(\rho - z) - y,\\
\frac{\partial z}{\partial t} &= xy - \beta z.
\end{aligned}
$$

We define the system function as a solution to the differential equations defined by coupled equations $f'(x, y, z)$ in (3.4). The solutions are obtained by a solver in $Matlab^{\circledR}$. We discretise the continuous time index t with $n + \Delta\ t = n + 1$. The system function f and the measurement function h is described by the following equations

$$
\begin{aligned}
f(x,y,z,n) &= \int_{n-1}^{n} f'(x,y,z)\,\mathrm{d}t + w_n, && w \sim \mathcal{N}(0,1), \\
h(x,y,z,n) &= f(x,y,z,t)^3 + v_n, && v \sim \mathcal{N}(0,1).
\end{aligned}
\tag{3.5}
$$

The Lorenz system is bimodal and symmetric around zero. The cubic measurement function allows the filter to distinguish between the positive and negative modes.



(a) x axis

(b) y axis

(c) z axis

Figure 3.3.: Lorenz attractor filtering results

4 Discussion and Conclusion

The multimodal filter and smoother presented in the chapter 2 show marked improvement in the filtering and smoothing performance of a non-linear estimator, as evidenced by the state estimation results for standard non-linear benchmark problems. The filter and smoother both benefit from the proposed closed form Gaussian mixture approximation and mixture reduction techniques. In the following we analyse some of the key points in this report. A novel approach to multi-modal projections and the proposed Gaussian mixture smoothing also open up new avenues for research and we briefly discuss future work based on the numerical results.

The Salient Points of This Thesis

In the following, we discuss the implications of the multi-modal assumption and consider some related work to motivate possible extensions of the algorithms proposed in this thesis.

Need of the Multi-Modal Filter. We can see from the predictive densities for the Univariate Non-linear Growth Model bench problem that, for a nonlinear transformation of a Gaussian can become multi-modal. Moreover, the filtering with multimodal beliefs is significantly superior to unimodal Gaussian belief systems in terms of Root Mean Squared Error and Predictive Log Likelihood as metrics. It is worthwhile to note that, the gain in performance of the filter is due to transformation a Gaussian to a Gaussian mixture, which is different than Gaussian sum approximation in (Anderson and Moore, 2005), where the Gaussian mixture components serve the purpose of reducing trace(Σ) and not necessarily to capture multi modality. However, we believe that the superior performance of the multi-modal filter can also be partially attributed to the fact that by replacing a Gaussian with multiple Gaussians with smaller variances, we reduce the trace(Σ).

Multi-modal Approximation. While the flexibility of multi-modal beliefs allow us to reduce approximation errors, it also presents a challenge to evaluate the parameters of Gaussian mixture components. We proposed a closed form solution to estimation of Gaussian mixture parameters. The approximation is based on representing Gaussian by a Gaussian mixture such that the mean and covariance of the Gaussian mixture is same the original Gaussian. We assume the number of Gaussians to be fixed and their weights to be equal. The filter exhibits better performance than the Gaussian (uni-modal) filters even with the predetermination of some parameters of the Gaussian mixture, hence, we expect the performance to improve even further if we estimate the weights and the number of mixture components online. We mark online and efficient estimation of Gaussian mixture as a future work on multi-modal approximations

Multi-Modal Filtering. The multi-modal belief propagation and the closed form solution to mixture estimation give us a fast and highly flexible framework to model our transition dynamics, the multi-modal filter (Kamthe et al., 2013) exploits this framework along with Gaussian sum approximations to produce a very efficient non-linear estimation scheme.

The efficiency implies the ability of filter to be consistent (true value is not an outlier with respect to estimated probability density) and at the same time being able to compute results in reasonable time. The filter is an attempt to break the mould that for a multi-modal belief system particle filters are standard solutions. Through our results we show that the multi-modal filter, not only reduces error in mean square sense it also does it with consistency. The solutions are also deterministic and, hence, reproducible and do not suffer from the degeneracy associated with the particle filters. A consistent state estimate would imply that any inference or control task carried out with estimated probability distribution would account for the true value.

Multi-Modal Smoothing. The Gaussian sum smoother proposed in Section 2.2 is based on the forward-backward filter. The Gaussian sum smoother is a backward pass for the Gaussian mixture based forward multi-modal filter. The backward pass significantly improves the estimation results in terms of mean square error as shown by the numerical experiments. The smoother is based on the Gaussian sum approximation, hence, it guarantees optimality as the number of terms in approximation tends to infinity. Similar argument cannot be established for any Gaussian smoother available in literature, to the best of our knowledge. The Gaussian sum approximation allows us to establish the Gaussian sum filter as an optimal Bayesian filter as the number of terms in the mixture tend to infinity (Anderson and Moore, 2005). We can establish a similar claim for the proposed Gaussian sum smoother. The proposed Gaussian sum smoother reduces to an RTS smoother for the unscented Kalman filter proposed in (Särkkä, 2008). However, from our initial experiments we observe that the forward pass needs to be consistent for the Gaussian sum smoother to be optimal, and inconsistent forward pass can result in a negative covariance matrices and a divergent smoothing estimate. The exceptional performance of the proposed smoother in terms of mean square error is due to consistent forward filtering by multi-modal filter.

Filtering and Smoothing in General. The multi-modal filter and smoother can be treated as generalisation of Unscented Kalman filter (Julier and Uhlmann, 2004) and RTS smoother (Rauch et al., 1965). Our key contributions are *mutli-modal approximations* and *Gaussian sum smoother* together they form an efficient non-linear state estimation tool.

Mixture Reduction. We compared various mixture reduction techniques available in the literature. We carried out a brief and non-comprehensive study to determine the suitable mixture reduction technique. The mixture reduction is crucial for a consistent estimate, and, hence, a thorough study is warranted to determine an optimal reduction scheme.

The multi-modal filter introduces a new framework to approximate predictive densities of non-linear transition functions. The proposed method demonstrates a superior performance in standard benchmark problems and in future we would investigate its feasibility in a real life example.

4.1 Conclusion

In this thesis, we presented the M-MF, a Gaussian mixture based multi-modal filter for state estimation in non-linear dynamical systems. Multi-modal densities are represented by Gaussian mixtures, whose parameters are computed in closed form. We demonstrated that the M-MF achieves superior performance compared to state-of-the-art state estimators and consistently captures the uncertainty in multi-modal densities.We proposed a Gaussian sum smoother for a non-linear dynamical system with a closed form approximation to optimal backward pass.

4.2 Future work

In future work, we will evaluate the significance of the scaling parameter α and its impact on higher moments of the approximations. We plan to introduce additional hyperparamters to build more flexible estimation scheme. The hyperparameters may, e.g. control the weight assigned to different Gaussians in a mixture, which is fixed at $1/M$ for the current scheme with M Gaussians.

The effect of the mixture reduction techniques also will be investigated to achieve better filter performance. Mixture reduction needs to be a robust procedure, i.e. it should be able to determine the Gaussian components whose elimination has least impact on approximation accuracy.

The proposed multi-modal smoother can in principle be used for inference in switching linear dynamical systems. We would compare the proposed smoother with “Expectation Correction" (Barber, 2006), the forward-backward smoother (Vo et al., 2012) and particle filter based smoothers (Lindsten and Schön, 2013).

A Gaussian Conditioning

Consider variables x and y, where f is a non-linear functions. We are interested in $p(y|x,z)$.

$$x \xrightarrow{f} y$$

the probabilities $p(x)$ and $p(y)$ are assumed to be Gaussian

$$\begin{aligned} p(x) &= \mathcal{N}(x|, \mu_x, \Sigma_x), \\ p(y) &= \mathcal{N}(y|, \mu_y, \Sigma_y), \end{aligned} \tag{A.1}$$

We assume $p(x,y)$ to be jointly Gaussian as well, i.e.

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim \mathcal{N}\left(\begin{matrix} x \\ y \end{matrix} \middle| \begin{matrix} \mu_x \\ \mu_y \end{matrix}, \begin{matrix} \Sigma_x & \Gamma \\ \Gamma^T & \Sigma_y \end{matrix} \right) \tag{A.2}$$

If we represent the conditional densities $p(y|x)$ and $p(x|y)$ as,

$$\begin{aligned} p(y|x) &= \mathcal{N}(y|\mu'_y, \Sigma'_y) \\ p(x|y) &= \mathcal{N}(y|\mu''_x, \Sigma''_x) \end{aligned} \tag{A.3}$$

then we can evaluate parameters for conditonal Gaussian (Petersen and Pedersen, 2008) as

$$\mu'_y = \mu_y + \Gamma^T \Sigma_x^{-1}(x - \mu_x), \tag{A.4}$$

$$\Sigma'_y = \Sigma_y - \Gamma^T \Sigma_x^{-1} \Gamma, \tag{A.5}$$

and

$$\mu''_x = \mu_x + \Gamma^T \Sigma_y^{-1}(y - \mu_y), \tag{A.6}$$

$$\Sigma''_x = \Sigma_x - \Gamma^T \Sigma_y^{-1} \Gamma. \tag{A.7}$$

The Equations (A.4) to (A.7) along with the Bayes' theorem form the basis of Kalman (Gaussian) filter derivation starting from the Bayesian filter described in Section 1.2.

List of Tables

List of Figures

Bibliography

Alspach, D. and Sorenson, H. (1972). Nonlinear Bayesian estimation using Gaussian sum approximations. *Automatic Control, IEEE Transactions on*, 17(4):439–448.

Alspach, D. L. and Sorensen, H. W. (1972). Nonlinear Bayesian Estimation using Gaussian Sum Approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448.

Anderson, B. D. O. and Moore, J. B. (2005). *Optimal Filtering*. Dover Publications, Mineola, NY, USA.

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

Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons.

Barber, D. (2006). Expectation Correction for Smoothed Inference in Switching Linear Dynamical Systems. *Journal of Machine Learning Research*, 7:2515–2540.

Beran, R. (1977). Minimum hellinger distance estimates for parametric models. *The Annals of Statistics*, 5(3):445–463.

Deisenroth, M. P. and Ohlsson, H. (2011). A General Perspective on Gaussian Filtering and Smoothing: Explaining Current and Deriving New Algorithms. In *Proceedings of the American Control Conference*.

Deisenroth, M. P., Turner, R., Huber, M., Hanebeck, U. D., and Rasmussen, C. E. (2012). Robust Filtering and Smoothing with Gaussian Processes. *IEEE Transactions on Automatic Control*, 57(7):1865–1871. doi:10.1109/TAC.2011.2179426.

Doucet, A., De Freitas, N., Gordon, N., et al. (2001). *Sequential Monte Carlo methods in practice*, volume 1. Springer New York.

Gustafsson, F. and Hendeby, G. (2012). Some relations between extended and unscented kalman filters. *Signal Processing, IEEE Transactions on*, 60(2):545–555.

Hilborn, R. (2000). *Chaos and nonlinear dynamics: an introduction for scientists and engineers*. oxford university press.

Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422.

Kailath, T., Sayed, A. H., and Hassibi, B. (2000). *Linear estimation*, volume 1. Prentice Hall New Jersey.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME — Journal of Basic Engineering*, 82(Series D):35–45.

Kampa, K., Hasanbelliu, E., and Principe, J. C. (2011). Closed-form cauchy-schwarz pdf divergence for mixture of gaussians. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2578–2585. IEEE.

Kamthe, S., Peters, J., and Deisenroth, M. P. (2013). Multi-modal filtering for non-linear estimation. Submitted to ICASSP 2014.

Kitagawa, G. (1994). The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother. *Annals of the Institute of Statistical Mathematics*, 46(4):605–623.

Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25.

Kotecha, J. H. and Djuric, P. M. (2003). Gaussian sum particle filtering. *Signal Processing, IEEE Transactions on*, 51(10):2602–2612.

Lindsten, F. and Schön, T. B. (2013). Backward simulation methods for monte carlo statistical inference. *Foundations and Trends® in Econometrics*, 5(1-2).

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141.

Luo, X., Moroz, I. M., and Hoteit, I. (2010). Scaled unscented transform Gaussian sum filter: Theory and application. *Physica D: Nonlinear Phenomena*, 239(10):684–701.

Owens, N., Harris, C., and Stennett, C. (2003). Hawk-eye tennis system. In *Visual Information Engineering, 2003. VIE 2003. International Conference on*, pages 182–185. IET.

Petersen, K. B. and Pedersen, M. S. (2008). The Matrix Cookbook. Version 20081110.

Rauch, H. E., Tung, F., and Striebel, C. T. (1965). Maximum Likelihood Estimates of Linear Dynamical Systems. *AIAA Journal*, 3:1445–1450.

Runnalls, A. R. (2007). Kullback-leibler approach to gaussian mixture reduction. *Aerospace and Electronic Systems, IEEE Transactions on*, 43(3):989–999.

Särkkä, S. (2008). Unscented Rauch-Tung-Striebel Smoother. *IEEE Transactions on Automatic Control*, 53(3):845–849.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. The MIT Press, Cambridge, MA, USA.

Van Der Merwe, R., Doucet, A., De Freitas, N., and Wan, E. (2000). The unscented particle filter. In *Advances in Neural Information Processing Systems*, pages 584–590.

Vo, B.-N., Vo, B.-T., and Mahler, R. P. (2012). Closed-form solutions to forward–backward smoothing. *Signal Processing, IEEE Transactions on*, 60(1):2–17.

Wan, E. A. and van der Merwe, R. (2000). The Unscented Kalman Filter for Nonlinear Estimation. In *Proceedings of Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control*, pages 153–158, Lake Louise, Alberta, Canada.

Williams, J. L. and Maybeck, P. S. (2003). Cost-function-based gaussian mixture reduction for target tracking. In *Proceedings of the Sixth International Conference of Information Fusion*, volume 2, pages 1047–1054.