11-18-2022

Splitting Gaussian Densities to Minimize Variance Along a Direction of Nonlinearity

Amit Kumar
Portland State University

Splitting Gaussian Densities to Minimize Variance Along a Direction of Nonlinearity

by

Amit Kumar

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science
in
Electrical and Computer Engineering

Thesis Committee:
James McNames, Chair
Martin Siderius
Eric Wan

Portland State University
2022

Abstract

Nonlinear functions of random vectors are frequently used in signal processing, and especially in state space tracking algorithms. Many of these algorithms require a way of estimating the probability density of the state vector at the output of the nonlinear function. Algorithms derived from Kalman Filter, such as Extended Kalman Filter and Unscented Kalman Filter, are popular choices for this, but they only estimate mean and covariance which may be insufficient to describe the non-Gaussian densities. On the other hand, Monte Carlo methods such as particle filters can be more capable but require much more computation. Gaussian mixture filters aim to strike a balance between these two approaches. They offer more flexibility than the filters in the Kalman Filter family by being able to approximate any smooth density arbitrarily well, and at the same time typically require far less computation than the Monte Carlo methods. The number of components in a Gaussian mixture is often chosen to balance the trade-off between computation and accuracy. When necessary, new components are typically created by splitting one of the components in a single direction. This work proposes a new method for determining the direction of split that minimizes the variance of the new components along the direction of nonlinearity. This results in more localized linear function approximation that generally improves accuracy. The proposed direction

of split is close to optimal, and performs better than popular alternatives in several examples detailed in this thesis.

to teachers everywhere

Acknowledgments

Whatever measure of success I have had on this work has been only possible because so many people have supported me at every step.

First of all, I am deeply thankful my advisor, Dr. James McNames. His patience, guidance, wealth of knowledge and enthusiasm has made all the time spent on this work immensely enriching, educational and enjoyable.

I also have the sincerest appreciation for the ECE faculty. Every signal processing course I took was, besides being enlightening, an amazing experience, a privilege, and a lesson in technical integrity and intellectual humility.

But the biggest debt of gratitude I owe is to my wife, Shefali. I have stolen away so much time and attention that should have been hers. She not only allowed me to do so, but enabled me willingly and heartily so that I could joyfully follow my passion in research. Also, my kids have my most affectionate and grateful thanks for letting me work on this thesis when it mattered the most.

Finally, I must also acknowledge that all the simulations, which form the foundation of this work, have benefited tremendously from myriad Python libraries and countless cups of coffee.

Table of Contents

List of Tables

List of Figures

1 Introduction

State space tracking is used in many applications to track the state vector of a stochastic dynamic system. Typically the state changes with time, and its exact value is not known. The information about the state is conveyed by a probability density. This work is in the context of discrete-time state space tracking where, at each time step, the true state vector changes based on a deterministic function referred to as the *state transition function*. Correspondingly, the probability density representing the state also needs to be updated at each time step. Let's express the state transition function as

$$y = f(x) \tag{1.1}$$

where x and y respectively denote the state vectors before and after the state transition. This work deals with the problem of *density propagation*, *i.e.*, the task of estimating the probability density of y, represented by $q(y)$, given the probability density of x, represented by $p(x)$ and a nonlinear state transition function $f(x)$. Terms *prior density* and *propagated density* are also used to refer to $p(x)$ and $q(y)$ respectively. The place and application of this work within the broader state space tracking framework is discussed in more detail in Chapter 5.

1.1 Current Practices

This section introduces some of the popular methods used for estimating propagated densities for nonlinear state transition functions. Firstly, it highlights the two most popular classes of techniques – the Kalman Filter family and the Monte Carlo methods – used for this purpose. Then it introduces the Gaussian mixture-based methods that attempt to strike a balance between the two. This establishes the context for the main contribution of this work.

1.1.1 Kalman Filter Family

Kalman Filter (KF) and its extensions [34], such as Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF), represent the state probably density by a Gaussian probability density function (PDF). When the state transition function $f(x)$ is linear or affine[1], *i.e.* $f(x) = ax + b$, the Kalman Filter provides an exact solution for the propagated state density, which also happens to be Gaussian. But, in practice, the state transition function $f(x)$ may be nonlinear in many applications. Estimating the propagated density becomes more difficult in those cases and one must use alternatives such as EKF or UKF [1, 21, 31]. Both of them estimate the mean and covariance of the propagated densities based on different linearized approximations of $f(x)$. EKF uses Taylor series expansion of the nonlinear func-

[1]This text does not differentiate between linear and affine functions for sake of conciseness. Throughout the thesis, the use of the term "linear function" is intended to include affine functions. Similarly, the use of the term "nonlinear function" excludes the affine functions. This is consistent with most sources on state space tracking.

tion to analytically create a linear approximation in order to be able to use the Kalman Filter update equations. UKF does not sacrifice the function nonlinearity, and applies unscented transform [19] on the sigma points of the input density to estimate the output mean and covariance. Both EKF and UKF are computationally efficient, but only partly characterize the true propagated density. Between EKF and UKF, UKF is more effective in dealing with nonlinearities [19,31]. But despite its remarkable simplicity and the effectiveness in handling many nonlinear state transition functions, some of UKF's limitations should also be noted. UKF only estimates the first and seconds moments of the density, which only fully characterize the density if the density is Gaussian. Consider, for example, a state transition function that produces the propagated density with more than one mode. In such cases, no single Gaussian PDF, including those obtained by UKF, might be be a good representation of the new state density. Additionally, even when the true propagated density has a single mode, it is possible for the function nonlinearity to be severe enough such that the true behavior is not captured well by the UKF update equations. Sometimes these limitations result in large estimation error.

1.1.2 Monte Carlo Techniques

Monte Carlo techniques, such as Particle Filters (PFs), are popular alternatives [5, 7, 13]. These represent the state density as a weighted random sample of state vectors, also referred to as *particles*. The ensemble of particles can represent a much broader range of densities than the Gaussian form. The representation also

generally converges to the true value asymptotically as the number of particles increases [17].

Monte Carlo techniques also have a few drawbacks that should be noted. Firstly, certain problems may need a very large number of particles, thus resulting in high computational cost. This often happens when the dimensionality of the state vector is high or when the state dynamics are such that the state evolution only depends on a small fraction of all particles. Another shortcoming of PFs is that while certain properties of $q(y)$, such as mean and covariance, may be estimated easily, certain other properties, such are number of modes, are difficult to estimate. This stems from the fact that Monte Carlo techniques lack a canonical form representation of the state density. So it is difficult to determine the exact value of the probability density for an arbitrary value of the state vector, and one must resort to approximations such as kernel smoothing [27] when that is needed.

1.1.3 Gaussian Mixture Models

Gaussian mixture models have the potential to provide estimates of the propagated density that are both more general than the EKF and UKF, but also more computationally efficient than PFs. They represent the state density, $\hat{p}(x)$, by a Gaussian mixture, *i.e.* a weighted sum of individual Gaussians distributions. This can be expressed as

$$\hat{p}(x) = \sum_{n=1}^{N} \alpha_n \mathcal{N}(x; \mu_n, P_n) \tag{1.2}$$

where α_n is the weight of the nth component, and $\mathcal{N}(x; \mu_n, P_n)$ is a Gaussian density with a mean vector of μ_n and a covariance matrix of P_n.

Thus, they possess a canonical form, but can also represent any smooth density, including densities with multiple modes [28], with arbitrary degree of accuracy given sufficient number of Gaussian components. If the mixture uses a large number of components, each with a small covariance such that $f(x)$ can be well approximated as a local linear function for that component, then the resulting density $q(y)$ can also be well approximated as a Gaussian mixture, and the mean and covariance of each component can be calculated using the EKF or UKF update equations [2]. If one of the mixture components spans over a large domain of the nonlinear function, the linear approximation may be significantly inaccurate which in turn can cause the estimate of the propagated density $q(y)$ to be inaccurate. The accuracy can be improved by subdividing the Gaussian component into multiple subcomponents where each of the subcomponent Gaussians has a smaller covariance than the original. This is illustrated using an example in Figure 1.1. However, processing additional components requires more computation, so the total number of components N has a significant influence on the trade-off between the computational cost and accuracy of the representation.

One of the major challenges with the Gaussian mixture methods is that having a fixed number of Gaussian components at all times during the tracking may not work well. At a new time step, the number of Gaussian components may need to be increased such that the covariance of each individual component gets small enough to deal with the nonlinearity effectively. On the other hand, since having a large

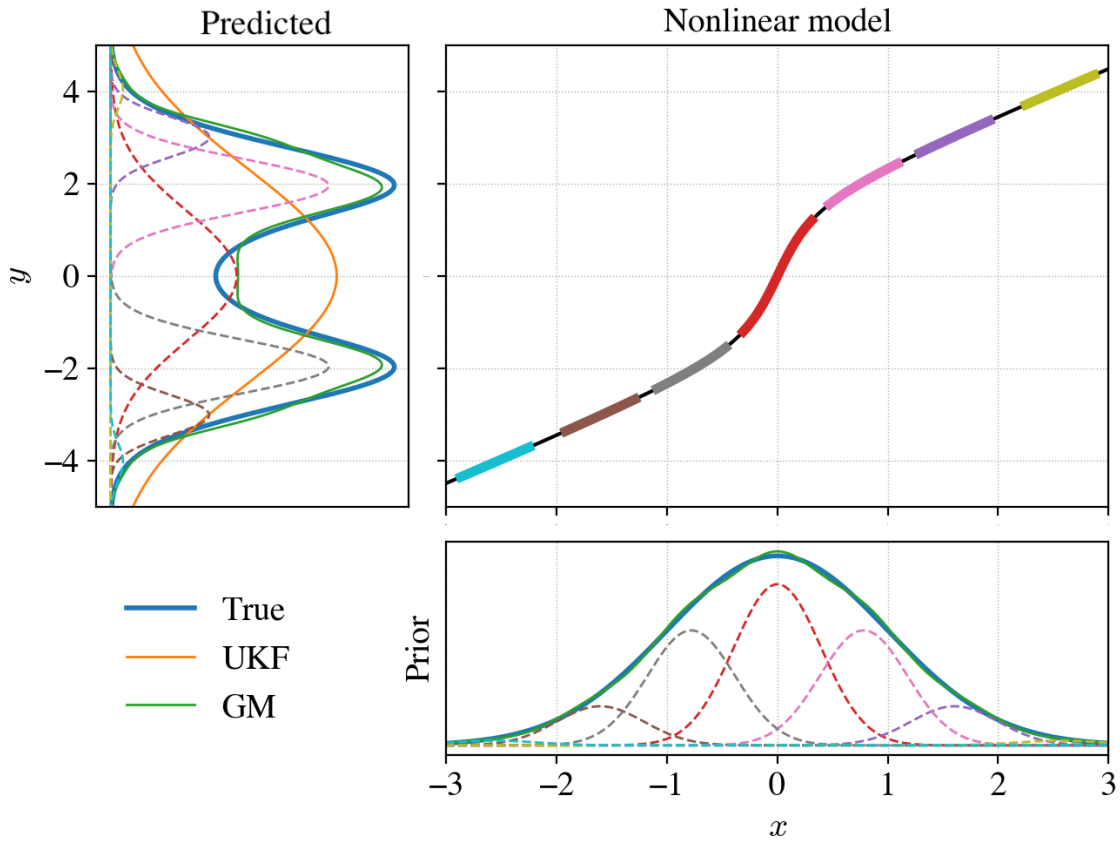

Figure 1.1: Density update for a univariate nonlinear function. The state update equation is $y = x + 2 \arctan 4x$, which is shown by the curve in the top-right subplot. The prior density consists of a single standard Gaussian component, shown in the bottom subplot. Its Gaussian mixture approximation consists of seven Gaussian subcomponents of smaller variance which are shown by dashed lines. The 1.5 sigma region for each of the Gaussian subcomponent is also highlighted in the top-right subplot. The propagated densities are shown in the left subplot. The Gaussian mixture update results in a much better approximation of the true propagated density compared to the single Gaussian. This is due to the nonlinear function appearing less nonlinear to each of the Gaussian mixture subcomponents with smaller variance.

number of components is computationally expensive, the number of components should also be reduced when possible. A common way to increase the number of components in a Gaussian mixture is to split one or more of the components into subcomponents. Similarly, one way to reduce the number of components in a Gaussian mixture is to merge two or more Gaussian components into a single Gaussian component.

1.2 Gaussian Split Literature Review

One of the most popular ways of splitting a multivariate Gaussian component is along the eigenvectors or principal axis of its covariance matrix. Zhang *et al.* [35], and Vishwajeet and Singla [32] present solutions for splitting a Gaussian component into two subcomponents along each of the eigenvectors of its covariance matrix. This can be trivially reduced to split only along a subset of the eigenvectors. Very often this subset is limited to just the principal axis as in [6, 16, 18, 23].

Terejanu [30], and Faubel and Klakow [11] attempt to split Gaussian components along the direction of state transition function nonlinearity. But they limit the direction of split to one of the eigenvectors for sake of simplicity and computational efficiency. Terejanu limits the search for the direction of nonlinearity to the set of eigenvectors of the covariance matrix. Faubel and Klakow identify the direction of nonlinearity without this constraint but later quantize that to the nearest eigenvector because, in their words, 'splitting a Gaussian distribution in an arbitrary direction Ψ turns out to be difficult unless Ψ coincides with one of

the principal axes of the covariance matrix'. Presumably the said 'difficulty' is in guaranteeing that the subcomponent covariance matrices resulting from the splitting formulae remain positive definite. A matrix must be positive definite to be a valid choice for the covariance matrix of a Gaussian distribution. It is analogous to the necessity that the variance of a univariate Gaussian distribution be positive. It is worth noting that the work presented in this thesis, by construction, always results in valid Gaussians subcomponents with positive definite covariance matrix.

Many authors such as Havlak and Campbell [14], Raitoharju *et al.* [25], and Vishwajeet and Singla [33] have suggested various methods for determining the direction of maximum nonlinearity in the state transition functions. They also provide methods of splitting Gaussian components along that direction.

A variation of splitting along the direction of maximum nonlinearity is proposed by Duník *et al.* in [9], where the split is performed along the direction of maximum 'non Gaussianity', which is defined as the normalized third-order central moment of the transformed variable [29].

1.3 Significance of This Work

This work deals with the problem of splitting a Gaussian component into multiple subcomponents in the context of the Gaussian mixture-based techniques. As mentioned previously, the split reduces the covariance of individual Gaussians in the mixture such that each component in the Gaussian sum is able to better deal with the nonlinear system dynamics.

As seen in Section 1.2, the most common way of performing the split is just along a single direction. The direction of split is an important design decision. It is common to choose the direction of split to be along the principal axis (DoPA) of the prior distribution as that is the direction of maximum spread for the distribution. However a better choice for the direction of split would also take into account the nonlinearity in the state transition function. For example, splitting the prior distribution along its principal axis would not yield any benefit if the function nonlinearity only existed in an orthogonal direction. If the direction of nonlinearity (DoNL) is known, it is common to perform the split along that direction. If it is not known, it can be estimated [9, 26]. This work shows that in the situations where the direction of nonlinearity is either known or can be estimated, an even better direction of split can be used. This thesis refers to it as the direction of minimum variance (DoMV).

1.4 Symbols and Notations

Table 1.1 is a consolidated list of important symbols and notations used in this thesis.

Table 1.1: Symbols and notations.

Symbol	Description
x, y	State, before and after, the nonlinear transition
$p(x), q(y)$	Prior and propagated densities (before and after the nonlinear transition)
N	Number of components in a Gaussian mixture
α_n, μ_n, P_n	Weight, mean and covariance of the nth Gaussian component in the prior density Gaussian mixture
μ'_n, P'_n	Mean and covariance of the nth Gaussian component in the propagated density Gaussian mixture
s	Vector representing a direction of split
u, s_{MV}	Vectors representing the direction of function nonlinearity (DoNL) and the direction of split for minimizing variance (DoMV)
K	Number of samples in the Monte Carlo simulation
$\hat{q}(y), \bar{q}(y)$	propagated density estimated using Gaussian mixture and Monte Carlo method respectively
δ, g_{ij}	Square grid spacing and grid points for numerically estimating KL divergence
w_{ij}	Two-dimensional histogram of the Monte Carlo samples

1.5 Thesis Outline

The remainder of the thesis is organized as follows. Chapter 2 describes and derives the proposed algorithm. Chapter 3 specifies the method used for evaluating the proposed algorithm. Chapter 4 presents and discusses the evaluation results. Finally, Chapter 6 summarizes the work and provides conclusions.

2 Algorithm

As seen in Section 1.1.3, the individual components of a multivariate Gaussian mixture distribution are often split into subcomponents to create a mixture consisting of Gaussians with smaller covariance. There are many ways of performing the split, but as seen in Section 1.2, most commonly it is performed just along a single direction, and when the direction of nonlinearity (DoNL) in the system transition function is known or can be estimated, the split is performed in the same direction.

This chapter firstly describes what it means to split a Gaussian distribution along a direction. It then discusses the intuition behind the belief that a better direction of split than DoNL must exist, and proceeds to derive it.

2.1 Splitting a Gaussian Density Along a Direction

Splitting a Gaussian component into subcomponents *along a direction* refers to the fact that the means of the new subcomponents are located on a straight line parallel to the direction of split.

Consider the case where the prior density, $p(x)$, has just one Gaussian component with mean μ and covariance P.

$$p(x) = \mathcal{N}(x; \mu, P). \tag{2.1}$$

It is split into a Gaussian mixture, $\hat{p}(x)$, represented as

$$\hat{p}(x) = \sum_{n=1}^{N} \alpha_n \mathcal{N}(x; \mu_n, P_n) \tag{2.2}$$

where α_n is the weight of the nth component and $\mathcal{N}(x; \mu_n, P_n)$ is a Gaussian density with a mean vector of μ_n and a covariance matrix of P_n. The direction of split is said to be along a vector s if the mean of the individual subcomponents after the split are related to the original mean by

$$\mu_n = \mu + m_n s \tag{2.3}$$

where m_n is a scalar.

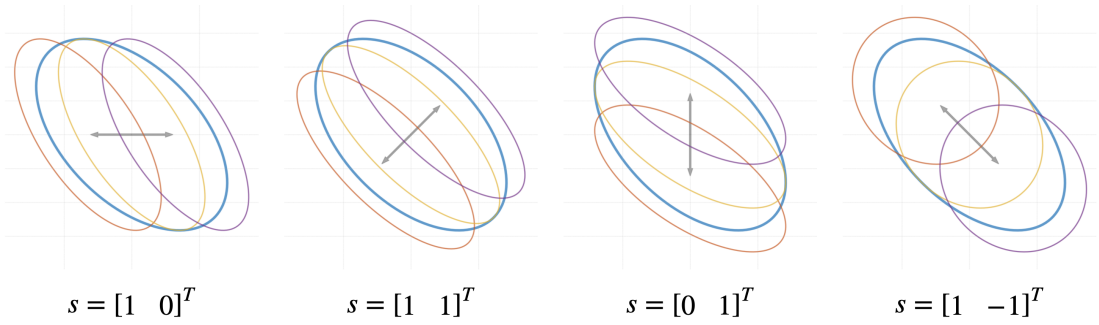

Figure 2.1: Examples of splitting a two-dimensional Gaussian density into three subcomponents along a direction. Each Gaussian density is represented by an ellipse outlining its two-sigma boundary. The blue ellipse with thick line represents the original Gaussian component, while other ellipses represent the split subcomponents. The direction of split is given by the vector s whose first and second components represent the horizontal and vertical directions respectively.

Limiting the split of a multivariate Gaussian to a single direction in this manner is desirable because then, through simple transformations, it can be reduced to a much simpler problem of splitting a standard univariate Gaussian. This is explained well by DeMars *et al.* in [6] as: 'The best way to think of this is to consider the principal directions of the covariance matrix (given by the eigenvectors of the covariance matrix). Then, in the coordinate system described by the principal directions, the multivariate Gaussian distribution becomes a product of univariate Gaussian distributions, which allows for the straightforward implementation of a univariate splitting technique to be applied to any one, several, or all of the elements in this product of univariate Gaussian distributions. While thinking of the

principal directions provides physical insight into the problem, it is not required for describing the general approach.'

2.1.1 Splitting Univariate Gaussian

Consider the case of splitting a standard univariate Gaussian, *i.e.*, $\mu = 0$ and $P = 1$ in (2.1). In general, the new Gaussian mixture density $\hat{p}(x)$ cannot exactly equal the original density $p(x)$, except in the trivial and uninteresting case where all the subcomponents in $\hat{p}(x)$ have the same mean and covariance as the original density. This presents an optimization problem with two conflicting objectives: producing subcomponents with small covariances *and* keeping the approximation error small. It is a common practice to let each of the subcomponents after the split have the same covariance to simplify the optimization process.

Representing the subcomponent covariance as $\bar{\sigma}^2$, DeMars *et al.* [6] frame the standard univariate splitting problem as minimizing the loss function

$$J = D\left(p(x) \| \hat{p}(x)\right) + \lambda \bar{\sigma}^2 \tag{2.4}$$

where λ is a design parameter representing the weight applied to the objective of minimizing covariance of the new subcomponents, and the divergence D is a measure of the approximation error in $\hat{p}(x)$ relative to $p(x)$. Kullback–Leibler divergence (KL divergence) and integrated squared error (ISE) are a couple of examples that could be used for such a measure. For the specific case of ISE, $D\left(p(x) \| \hat{p}(x)\right)$ can be found in closed form. DeMars *et al.* use this to solve the

optimization problem with $N = 3$ and $\lambda = 0.001$ in [6]. The results from the optimization are summarized in the table below.

Table 2.1: DeMars *et al.*'s three-component splitting library.

n	α_n	$\bar{m}_n$	$\bar{\sigma}^2$
1	0.2252	-1.0575	0.6716
2	0.5496	0	0.6716
3	0.2252	1.0575	0.6716

Note that the optimization for the standard univariate split can be performed offline. The optimized parameter values listed in the table can be saved for reuse during run-time. Thus, the existence of the closed-form solution for ISE is only a minor convenience. For example, one could use KL divergence for D and estimate it numerically for the optimization.

2.1.2 Splitting Multivariate Gaussian

DeMars *et al.* also present a method to split a multivariate Gaussian along an arbitrary direction in [6]. It involves finding a square-root of P, say S, that has the desired split direction s as one of its columns. Without loss of generality, let's say that the ith column, s_i, represents the desired direction of split, *i.e.*,

$$s_i = ks \tag{2.5}$$

where k is a scalar and

$$S = [s_1, \cdots, s_i, \cdots, s_L] \tag{2.6}$$

$$P = SS^T \tag{2.7}$$

where L is the dimensionality of the state vector. The means and covariance of the Gaussian subcomponents after the split along s_i, or equivalently s, are given by

$$\mu_n = \mu + \bar{m}_n s_i \tag{2.8}$$

$$\bar{S} = [s_1, \cdots, \bar{\sigma} s_i, \cdots, s_L] \tag{2.9}$$

$$P_n = \bar{S}\bar{S}^T \qquad \text{for } 1 \leq n \leq N \tag{2.10}$$

where the values of m_n and $\bar{\sigma}^2$ are listed in Table 2.1.

2.1.3 Alternative Form for Splitting Multivariate Gaussian

The following derives an alternative set of equations for splitting a multivariate Gaussian. It is functionally equivalent to the method outlined in the previous section, but does not require finding any square-root of the covariance matrix.

Note the relationship

$$\det(P_n) = \bar{\sigma}^2 \det(P) \tag{2.11}$$

since

$$\det(P) = |\det(S)|^2,$$

$$\text{and,} \qquad \det(P_n) = |\det(\bar{S})|^2 = \bar{\sigma}^2 |\det(S)|^2.$$

Also note that (2.7) and (2.10) can be expanded as

$$P = s_1 s_1^T + \cdots + s_i s_i^T + \cdots + s_L s_L^T,$$

$$\text{and,} \qquad P_n = s_1 s_1^T + \cdots + \bar{\sigma}^2 s_i s_i^T + \cdots + s_L s_L^T.$$

Taking their difference,

$$P - P_n = (1 - \bar{\sigma}^2) s_i s_i^T. \tag{2.12}$$

Substitution s_i by ks from (2.5), and rearranging,

$$P_n = P - k^2 (1 - \bar{\sigma}^2) s s^T. \tag{2.13}$$

Taking determinant on both sides and using the matrix determinant lemma allows solving for k as follows:

$$\det(P_n) = \det(P - k^2 (1 - \bar{\sigma}^2) s s^T),$$

$$\bar{\sigma}^2 \det(P) = \det(P) \left(1 - k^2 (1 - \bar{\sigma}^2) s^T P^{-1} s\right),$$

$$\bar{\sigma}^2 = 1 - k^2 (1 - \bar{\sigma}^2) s^T P^{-1} s.$$

Thus,

$$k = \frac{1}{\sqrt{s^T P^{-1} s}} \tag{2.14}$$

Substituting the value of k from (2.14) and s_i from (2.5) into (2.8) and (2.13) results in the following set of equations for computing the mean and covariance of the Gaussian subcomponents after the split.

$$\mu_n = \mu + \bar{m}_n \frac{s}{\sqrt{s^T P^{-1} s}} \tag{2.15}$$

$$P_n = P - (1 - \bar{\sigma}^2) \frac{s s^T}{s^T P^{-1} s} \qquad \text{for } 1 \leq n \leq N \tag{2.16}$$

This produces exactly the same set of subcomponents as the method outlined in the previous section. But this novel form avoids having to find any square-root of P, much less a square-root containing the desired split direction as one of its columns.

2.2 Direction of Split for Minimum Variance (DoMV)

The motivation behind splitting a Gaussian component of the prior distribution into further subcomponents is to make the linear approximation of the nonlinear system transition function more accurate over the domain of each of the subcomponents. The same logic dictates that when the function nonlinearity is along a particular direction, say DoNL, one should choose the direction of split that would minimize the spread or variance of the split components along the DoNL.

Note that splitting a Gaussian along the DoNL does not necessarily minimize the variance of the resulting subcomponents along DoNL. The following example illustrates this. Let's assume, without loss of generality, that the DoNL is $[1\ 0]^T$ for a two-dimensional state vector. In other words, the coordinate system is such that the the DoNL is along the x-axis. Consider a Gaussian component with mean $[0\ 0]^T$ and covariance

$$\begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}.$$

Splitting this Gaussian component along any general direction results in multiple subcomponents that are offset from each other along that direction. Using the method explained in Section 2.1, the split produces three subcomponents of equal covariance. Fig. 2.2 shows the variance of these subcomponents along the DoNL for all possible angles of split. It confirms that splitting along DoNL, which is 0° in this example, may not produce subcomponents with the minimum variance along DoNL. The angle of split that does produce minimum variance along DoNL is defined as DoMV in this work. Note that DoMV is also different from the principal axis, which is shown by the green marker.

Fig. 2.3 compares splitting along DoNL with splitting along DoMV for the same example in more detail. The two directions of split result in different subcomponents. Marginals of the subcomponents along the x-axis, which is the DoNL, show how the two directions of split result in two different approximations of the original marginal. As expected, the subcomponents for split along DoMV have smaller spread along DoNL than the subcomponents for split along DoNL.

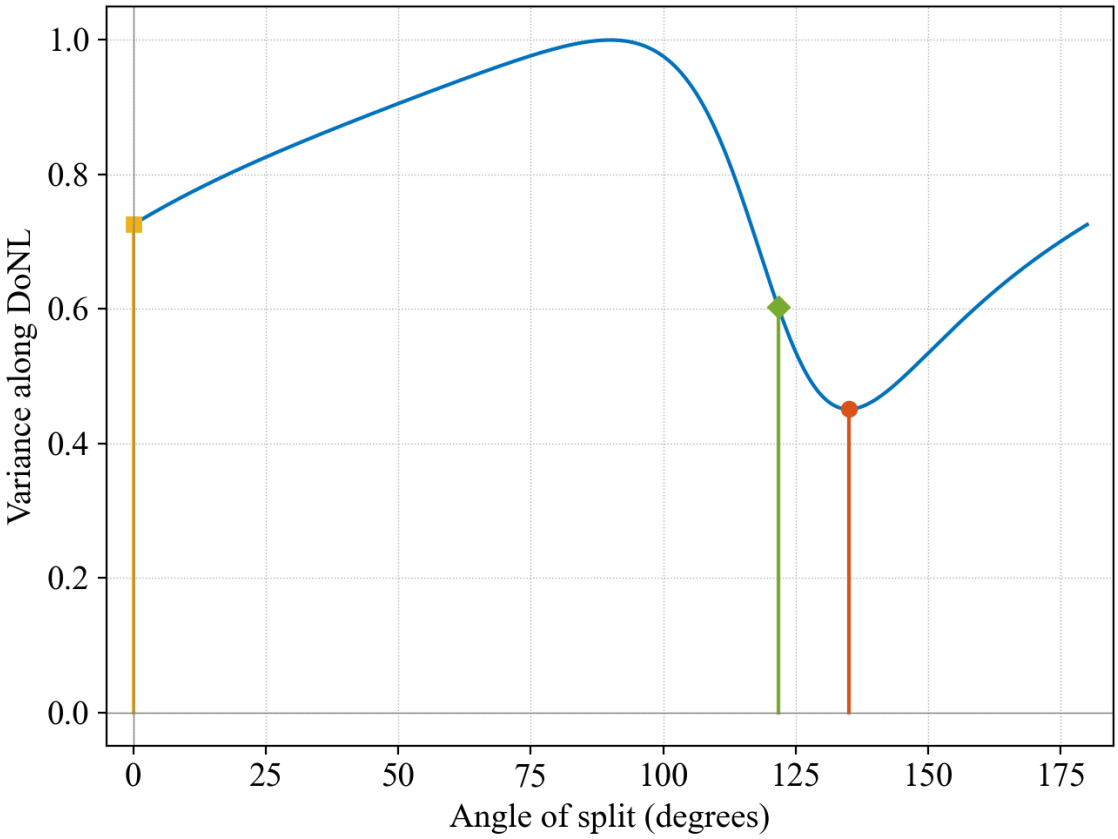

Figure 2.2: Example of subcomponent variance along DoNL for various angles of split. The yellow square represents the case when the split is along DoNL, *i.e.*, 0°. The red circle marks the minimum of the curve, and the corresponding angle of split is defined as DoMV. Note that DoMV is also different from the principal axis of the original component, which is annotated with the green line and a diamond marker.

Let vector u represent the DoNL, and s_{MV} represent the DoMV, *i.e.*, the direction of split that would minimize the variance of the split components along u. This can be expressed as

$$s_{\text{MV}} = \underset{s}{\operatorname{argmin}}\, u^T P_n u \tag{2.17}$$

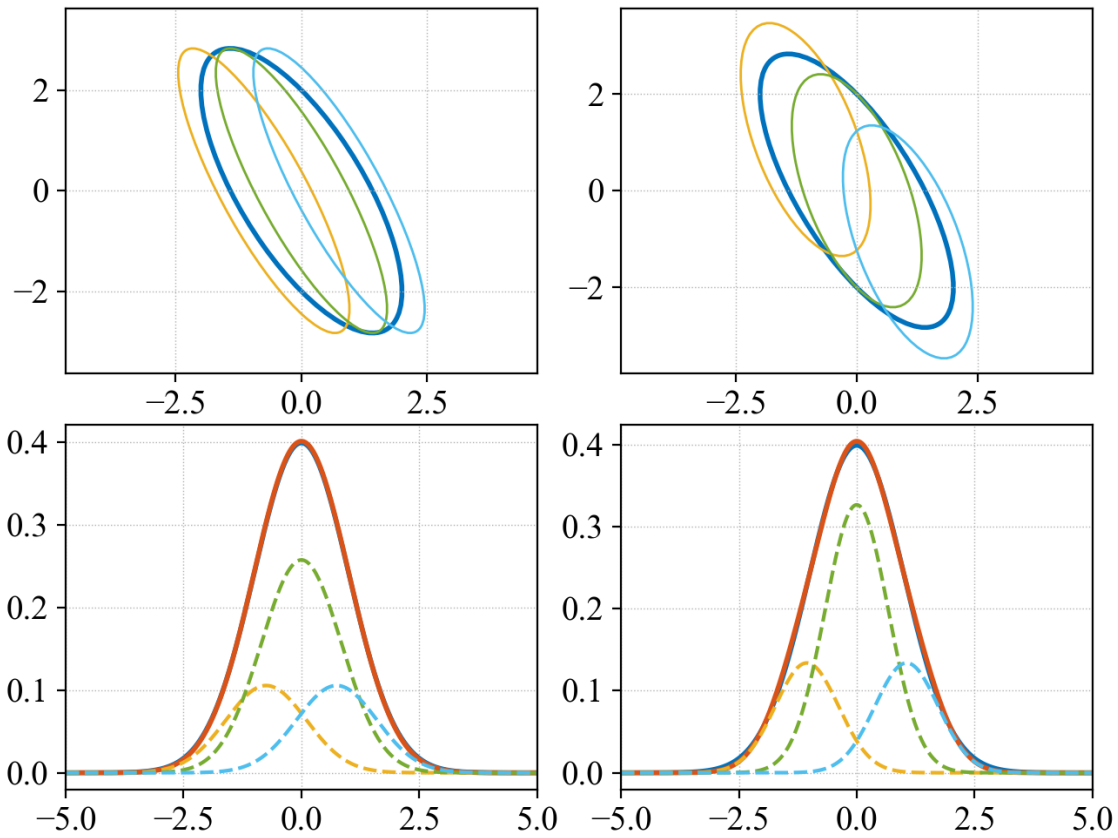

Figure 2.3: Example Gaussian component split along DoNL and DoMV. The top-left subplot shows the case when the original Gaussian component (thick blue ellipse) is split along DoNL into three subcomponents (thinner ellipses of various colors). Since the DoNL is 0°, the subcomponents are spread along the x-axis. The top-right subplot shows the same for the split along DoMV. The bottom two subplots show the corresponding marginal distributions along the DoNL. The marginals have smaller spread when the split is along DoMV.

where P_n is a function of s as shown in (2.16). Thus,

$$s_{\text{MV}} = \underset{s}{\operatorname{argmin}}\, u^T \left(P - (1 - \bar{\sigma}^2) \frac{s s^T}{s^T P^{-1} s} \right) u.$$

Since the minimization is over s, the terms that do not depend on s can be safely

ignored. So,

$$\begin{aligned} s_{\mathrm{MV}} &= \underset{s}{\operatorname{argmin}} \frac{-u^T s s^T u}{s^T P^{-1} s} \\ &= \underset{s}{\operatorname{argmax}} \frac{(u^T s)^2}{s^T P^{-1} s} \end{aligned} \tag{2.18}$$

Using any square-root Q of the positive definite matrix P such that $P = QQ^T$, this can be rewritten as

$$s_{\mathrm{MV}} = \underset{s}{\operatorname{argmax}} \frac{(u^T s)^2}{s^T Q^{-T} Q^{-1} s} \tag{2.19}$$

where the superscript $^{-T}$ represents the operations of both inverse and transpose. Using substitution $v = Q^{-1}s$, or equivalently, $s = Qv$, this can be rewritten as

$$s_{\mathrm{MV}} = Q \underset{v}{\operatorname{argmax}} \frac{(u^T Q v)^2}{v^T v} \tag{2.20}$$

Both the numerator and the denominator of the term being maximized are proportional to the square of the magnitude of v. Hence, the maximization only depends on the direction of v and not the magnitude. Thus, the maximization happens when vectors v and $Q^T u$ have the same direction. In other words,

$$\underset{v}{\operatorname{argmax}} \frac{(u^T Q v)^2}{v^T v} = c Q^T u \tag{2.21}$$

where c is any scalar. For simplicity and without loss of generality, let's choose c

to be unity. Substituting this back in (2.20) results in

$$s_{\mathrm{MV}} = QQ^T u = Pu. \tag{2.22}$$

Thus, the variance of the split subcomponents is minimized along a given direction u when the original Gaussian component is split along the direction given by the vector Pu. This is a crucial result. It shows how to compute the DoMV when DoNL is known. It also shows that, in general, DoMV and DoNL are different. The only case when they are the same is when DoNL is along one of the eigenvectors of the covariance matrix P. Furthermore, when the split is performed along the DoMV, the variance of the split components along DoNL, *i.e.* u, is

$$\begin{aligned} & u^T P_n u \\ &= u^T \left(P - (1-\bar{\sigma}^2) \frac{s_{\mathrm{MV}} s_{\mathrm{MV}}^T}{s_{\mathrm{MV}}^T P^{-1} s_{\mathrm{MV}}} \right) u \\ &= u^T Pu - (1-\bar{\sigma}^2) \frac{u^T Puu^T P^T u}{u^T P^T P^{-1} Pu} \\ &= \bar{\sigma}^2 u^T Pu \end{aligned} \tag{2.23}$$

Thus, splitting along DoMV reduces the variance of the individual Gaussian components along DoNL by a factor of $\bar{\sigma}^2$. This is the most reduction that could be expected since the underlying optimized univariate Gaussian split library in Section 2.1.1 reduces the univariate variance by the same factor. It is comforting to see that the maximum variance reduction happens in the direction of nonlinearity. The same aspect is also reflected by the fact the determinant of the covariance

matrix after the split shrinks by a factor of $\bar{\sigma}^2$ as seen is (2.11).

3 Assessment

The general goal of state space tracking methods is to estimate the probability density of a state vector. Therefore, the most direct way of evaluating the performance of these methods is to compare the estimated probability density with the true density. The same applies to this work, which proposes a new direction of split, DoMV, for Gaussian mixture filters, which is also a state space tracking method. Since a different direction of split results in a different estimate of the propagated density, ascertaining whether a particular direction of split is a better choice than another, requires determining which of the two resulting density estimates is a better approximation of the true density. Hence, assessing the performance of the proposed scheme requires comparing the resulting propagated density with the true propagated density.

3.1 Measure of Performance

There are many different established measures of dissimilarity between two probability densities that could be used to quantify the approximation error in the propagated density. Kullback–Leibler divergence, or KL divergence, is one such widely used measure. Representing the true propagated density by $q(y)$, its KL

divergence relative to an approximation, $\hat{q}(y)$, is defined as

$$D_{\mathrm{KL}}(q\|\hat{q}) = \int_{-\infty}^{\infty} q(y)\left(\log q(y) - \log \hat{q}(y)\right) dy. \tag{3.1}$$

Like other measures of distance, KL divergence is always non-negative and is zero only when the two densities are identical. It is noteworthy that KL divergence is not symmetric and the measure depends on the order of the two arguments.

Another measure of dissimilarity between two probability densities is integrated squared error (ISE) distance, defined as

$$D_{\mathrm{ISE}}(q\|\hat{q}) = \int_{-\infty}^{\infty} |q(y) - \hat{q}(y)|^2 dy. \tag{3.2}$$

Unlike KL divergence, ISE is a symmetric metic, *i.e.*

$$D_{\mathrm{ISE}}(q\|\hat{q}) = D_{\mathrm{ISE}}(\hat{q}\|q). \tag{3.3}$$

Yet another popular choice for comparing univariate densities is the Kolmogorov–Smirnov distance, which is the maximum difference in the two *cumulative* densities. Representing the cumulative densities for $q(y)$ and $\hat{q}(y)$ as $Q(y)$ and $\hat{Q}(y)$ respectively, the Kolmogorov–Smirnov distance can be exprerssed as

$$D_{\mathrm{KS}}(q\|\hat{q}) = \max_{y} \left(|Q(y) - \hat{Q}(y)| \right). \tag{3.4}$$

But computing the KS distance for multivariate densities is significantly more

cumbersome than KL divergence and ISE distance because while the probability density for the Gaussian mixtures for any y can be computed readily in closed form, their cumulative densities must be estimated numerically.

This work uses both KL divergence and ISE distance to compare the true and propagated densities. This choice differs significantly from the most commonly encountered measure of performance in state space tracking literature, which happens to be the mean squared error (MSE) between the true state and the mean of the estimated state density. MSE is much simpler to compute than KL divergence. It also may be the only practical option when the values that the underlying state vector takes are known but the true posterior state probability density is unknown. A better MSE metric would be based on the mean of the true density instead of the value that the true state takes. However, in either event, the MSE is an oversimplified measure of performance for a density estimator, as it only depends on the mean, *i.e.* the first moment, of the estimated density. For example, consider the case of a bimodal estimated density with the two modes well-separated from each other. In this case, the mean of the density would be at an extremely unlikely value for the true state to take, and yet that is what the MSE metric would be based on.

In contrast, KL divergence and ISE distance compare the totality of the two densities. Hence, they are more comprehensive than MSE. It is reflected in the fact that while the MSE metric requires averaging over a number of time-steps to be meaningful, comparison between a propagated density and the true density is a valid, and often revealing, exercise at each time-step.

Section 3.2.4 describes in detail the method for estimating KL divergence and ISE distance of the true density relative to an estimated density in form of a Gaussian mixture.

3.2 Simulation Setup

Problems of uncertainty propagation provide good opportunities for examining the effects of different directions of split. In these problems, a random variable x with known probability density $p(x)$ is transformed by a nonlinear state transition function, as in (1.1), and the algorithm is tasked with predicting $q(y)$, the probability density of the output y.

3.2.1 Example State Transition Functions

The evaluation uses two different examples to probe the merits of splitting along DoMV. In both examples, the dimension of the state vectors is limited to two to simplify visualizations of the propagated densities. Since every direction in a two-dimensional space can be represented by a scalar angle, this also allows clear presentation of the results comparing the DoMV to all other possible directions of split. Furthermore, both examples use functions where the DoNL is known. In practice, there are a variety of methods that can be used to estimate the DoNL [9, 26].

The first example uses a function that converts two-dimensional polar coordinates to two-dimensional Cartesian coordinates. This is a nonlinear trans-

formation and is often encountered when the system dynamics include angular motion in Cartesian coordinates. This was analyzed previously by Duník *et al.* in [9]. The second example uses the arctan function to produce nonlinearity. It is a smooth transformation and yet produces bimodal output density even when the input is Gaussian. As mentioned earlier, both examples are limited to two-dimensional state vectors so that the probability densities and evaluation results can be presented clearly using two-dimensional plots. The Polar-to-Cartesian function transforms the Gaussian-distributed input into non-Gaussian distributed output, whereas the Arctan function output distribution is bimodal as well as non-Gaussian. Both of these problems are well suited for Gaussian mixture models. Table 3.1 lists the details of the chosen examples.

Table 3.1: Examples with nonlinear state transition functions.

Feature	**Polar-to-Cartesian Example**	**Arctan Example**
$y(0)^{\dagger}$	$x(0)\cos(x(1))$	$x(0)$
$y(1)^{\dagger}$	$x(0)\sin(x(1))$	$x(1)+2\arctan(4x(1))$
Prior mean, μ	$\begin{bmatrix} 2 \\ \frac{\pi}{4} \end{bmatrix}$	$\begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}$
Prior covariance, P	$\begin{bmatrix} 0.2 & 0.2 \\ 0.2 & \frac{\pi}{9} \end{bmatrix}$	$\begin{bmatrix} 1.0 & 0.8 \\ 0.8 & 1.1 \end{bmatrix}$

$\dagger$ $x = [x(0), x(1)]^T$ and $y = [y(0), y(1)]^T$ denote the input and output state vectors respectively.

It is clear that, for both functions, the nonlinearity exists only along the second input dimension $x(1)$. So their DoNL is 90° or, equivalently, along the vector $u = [0, 1]^T$.

3.2.2 Reference for the True Propagated Density

The true propagated density can be expressed as

$$q(y) = \int_x \Pr(y|x)\, p(x) dx. \tag{3.5}$$

where $\Pr(y|x)$ is the probability of the function output being y for the input x. Unfortunately, in general, it is not possible to compute the above integral in closed form for a nonlinear function, f. Hence, obtaining a reference for the true propagated density requires numerical methods. Two types of numerical methods are commonly used for this purpose. The first type consists of deterministic numerical integration performed over a finitely small grid. The computational complexity for numerical integration grows exponentially with the dimension of the state. The second type consists of Monte Carlo methods which utilize a number of random state vectors drawn from the original distribution. They are computationally more efficient than numerical integration for multidimensional states. Hence, this evaluation process uses a Monte Carlo method to obtain the reference for the true propagated density.

A key idea used in Monte Carlo methods is the ability to represent a probability density by a collection of state vectors. For example, let $\{x_k\}_{k=1}^{K}$ represent the set of K random state vectors where each of them is drawn independently from density $p(x)$. Let's also say that we are interested in the expected value of a property $h(x)$ of the state vectors. The true expectation is given by

$$\mathbb{E}(h(x)) = \int_x h(x)p(x)dx. \tag{3.6}$$

The Monte Carlo estimator for the same expectation is

$$\hat{\mathbb{E}}(h(x)) = \frac{1}{K}\sum_{k=1}^{K} h(x_k). \tag{3.7}$$

The bias of this estimator, *i.e.*, the difference between the expected value of the estimator and the true value, is

$$\begin{aligned}\mathbb{E}\left(\hat{\mathbb{E}}\left(h(x)\right)\right) - \mathbb{E}\left(h(x)\right) &= \mathbb{E}\left(\frac{1}{K}\sum_{k=1}^{K} h(x_k)\right) - \mathbb{E}\left(h(x)\right) \\ &= \left(\frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\left(h(x_k)\right)\right) - \mathbb{E}\left(h(x)\right).\end{aligned}$$

Since each x_k is independently drawn, $\mathbb{E}\left(h(x_k)\right) = \mathbb{E}\left(h(x)\right)$, and the above can be reduced to

$$\begin{aligned}\mathbb{E}\left(\hat{\mathbb{E}}\left(h(x)\right)\right) - \mathbb{E}\left(h(x)\right) &= \left(\frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\left(h(x)\right)\right) - \mathbb{E}\left(h(x)\right) \\ &= \mathbb{E}\left(h(x)\right) - \mathbb{E}\left(h(x)\right) \\ &= 0. \qquad (3.8)\end{aligned}$$

Thus, the Monte Carlo estimator is unbiased. As a specific example, setting $h(x) = x$ in (3.6) and (3.7) suggests that the average of the Monte Carlo samples is an unbiased estimator of the mean of the density. This can also be extended to higher order moments of the density $p(x)$. It is also evident from (3.7) that, as K increases, the variance of the estimator decreases. In fact, it asymptotically converges to the true value.

As the state transition function $f(x)$ is applied to the input x with density $p(x)$, let the output be y with density $q(y)$. The expected value of any property of

the output y can be obtained as

$$\mathbb{E}\left(h(y)\right) = \int_y h(y)q(y)dy. \tag{3.9}$$

Expressing this in terms of the original random variable x,

$$\mathbb{E}\left(h(y)\right) = \int_x h(f(x))p(x)dx \tag{3.10}$$

Using (3.7), an unbiased estimator for the above is

$$\hat{\mathbb{E}}\left(h(y)\right) = \frac{1}{K}\sum_{k=1}^{K} h(f(x_k)).$$

Finally, defining $y_k = f(x_k)$, this can rewritten as

$$\hat{\mathbb{E}}\left(h(y)\right) = \frac{1}{K}\sum_{k=1}^{K} h(y_k). \tag{3.11}$$

Thus, the properties of the output density $q(y)$ can be estimated using a set of state vectors $\{y_k\}_{k=1}^{K}$ where $y_k = f(x_k)$. In other words $\{y_k\}_{k=1}^{K}$ is the Monte Carlo representation of the true propagated density.

For the purposes of the simulations that follow, K is set to 10^6. This value was picked experimentally to be large enough such that the resulting variance in the Monte Carlo estimates are negligibly small.

3.2.3 Gaussian Mixture Density

The single Gaussian component of prior density is split into a Gaussian mixture of three Gaussian subcomponents by following the method outlined by (1.2), (2.15), (2.16), and Table 2.1. Let α_n, μ_n, and P_n respectively represent the weight, mean and covariance of the nth subcomponent. Let L be the dimension of the state vector.

Each of the Gaussian subcomponents is updated independently using the unscented transform, which is outlined below.

1. Find a sqare-root S_n or P_n such that $S_n S_n^T = P_n$. Let $S_n^{(i)}$ represent the ith column of S_n.

2. Create a set of $2L+1$ sigma vectors $\{\chi_i\}_{i=1}^{2L+1}$ and a corresponding set of weights $\{w_i\}_{i=1}^{2L+1}$, such that

$$\chi_1 = \mu_n, \tag{3.12}$$

$$w_1 = W, \tag{3.13}$$

and for $i = 1, 2, \cdots, L$,

$$\chi_{2i} = \mu_n + \gamma S_n^{(i)}, \tag{3.14}$$

$$\chi_{2i+1} = \mu_n - \gamma S_n^{(i)}, \tag{3.15}$$

$$w_{2i} = w_{2i+1} = \frac{1-W}{2L}, \tag{3.16}$$

where W and γ are design parameters. There are many different strategies for assigning values to these parameters [10, 31]. One of the the simplest strategy is to have $\gamma = \sqrt{L}$ and $W = 0$. This is also what the simulations in this work use.

3. Create a new set of sigma vectors $\{\chi'_i\}_{i=1}^{2L+1}$ such that $\chi'_i = f(\chi_i)$.

4. Compute the mean and covariance of output subcomponent as

$$\mu'_n = \sum_{i=1}^{2L+1} w_i \chi'_i, \tag{3.17}$$

$$P'_n = \sum_{i=1}^{2L+1} w_i (\chi'_i - \mu'_n)(\chi'_i - \mu'_n)^T. \tag{3.18}$$

Note that the weights α_n are not affected by the update equations above.

The mixture of updated Gaussian subcomponents provides an estimate of the true propagated density $q(y)$ and can be expressed as

$$\hat{q}(y) = \sum_{n=1}^{N} \alpha_n \mathcal{N}(y; \mu'_n, P'_n). \tag{3.19}$$

This process of updating the Gaussian mixture model is repeated for angles of split along DoMV, DoNL and all directions from 0 to 180 degrees in 1 degree increment as part of a grid-search to numerically determine the best direction of split.

3.2.4 Estimating KL Divergence and ISE Distance

As discussed earlier, the KL divergence and the ISE distance of the reference density relative to the Gaussian mixture approximation act as measures of performance for the Gaussian mixture method. As seen in (3.1) and (3.2), computations of KL divergence and ISE distance require evaluation of $q(y)$ and $\hat{q}(y)$ for general y. This is trivial for $\hat{q}(y)$ as it can be computed in closed form for a mixture of Gaussians. But the true propagated density $q(y)$ is represented just by a set of samples $\{y_k\}_{k=1}^K$, which does not delineate an obvious way of evaluating $q(y)$ directly. Instead, it must use a numerical approach, such as outlined below, to estimate the value of the true density $q(y)$.

Let g_{ij}, where i and j are signed integers, represent a point on a two-dimensional square grid on the plane of the state vector y. So,

$$g_{ij} = [i\delta, j\delta]^T \tag{3.20}$$

where δ is the grid spacing. Let w_{ij} represent the number of samples from the set $\{y_k\}_{k=1}^K$ that lie within a square of area δ^2 centered at g_{ij}. So the ratio $\frac{w_{ij}}{K}$ is an estimate of the probability mass within that area. Note that the set $\{w_{ij}\}_{ij}$ for all i, j represents a two-dimensional histogram of the Monte Carlo samples. Thus, the normalized histogram provides a way to get an approximation of the true density. Let's represent this as

$$\bar{q}(g_{ij}) = \frac{w_{ij}}{\delta^2 K} \tag{3.21}$$

where K has been previously defined as the total number of Monte Carlo samples. As seen earlier for other Monte Carlo estimates, this too is an unbiased estimate and converges to the true value asymptotically as K increases. Using this estimate of the true propagated density, the KL divergence can be numerically estimated using the summation

$$D_{\mathrm{KL}}(q\|\hat{q}) \approx \delta^2 \sum_{i,j} \bar{q}(g_{ij}) \left(\log \bar{q}(g_{ij}) - \log \hat{q}(g_{ij})\right). \tag{3.22}$$

Similarly, the ISE distance can be numerically estimated by

$$D_{\mathrm{ISE}}(q\|\hat{q}) \approx \delta^2 \sum_{i,j} |\bar{q}(g_{ij}) - \hat{q}(g_{ij})|^2. \tag{3.23}$$

4 Results and Discussion

As detailed in Chapter 3, the evaluation process consists of three major stages:

1. Representation of the prior density by
   (a) drawing Monte Carlo samples from the true prior density, and
   (b) creating Gaussian mixture approximation of the prior density by splitting.
2. Computation of propagated density by
   (a) propagating the Monte Carlo samples through the nonlinear state transition function to produce the reference true propagated density, and
   (b) using the UKF technique to estimate the propagated density for the Gaussian mixtures.
3. Estimation of the KL divergence of the reference density relative to the Gaussian mixture estimates.
4. Estimation of the ISE distance between the reference density and the Gaussian mixture estimates.

The results from each of the four stages, for each of the two examples, are presented in the following eight plots. Fig. 4.1 - Fig. 4.4 present the results for the example

with Polar-to-Cartesian nonlinearity. Similarly, Fig. 4.5 - Fig. 4.8 present the results for the example with Arctan nonlinearity.

Fig. 4.1, and similarly Fig. 4.5, shows the prior distributions, along with the Gaussian mixture approximation when the split performed along the DoMV for the two examples. One thousand of the total one million Monte Carlo samples drawn from the prior distribution are also shown. The small subset is sufficient for visualization of the density, and at the same time does not clutter the plots.

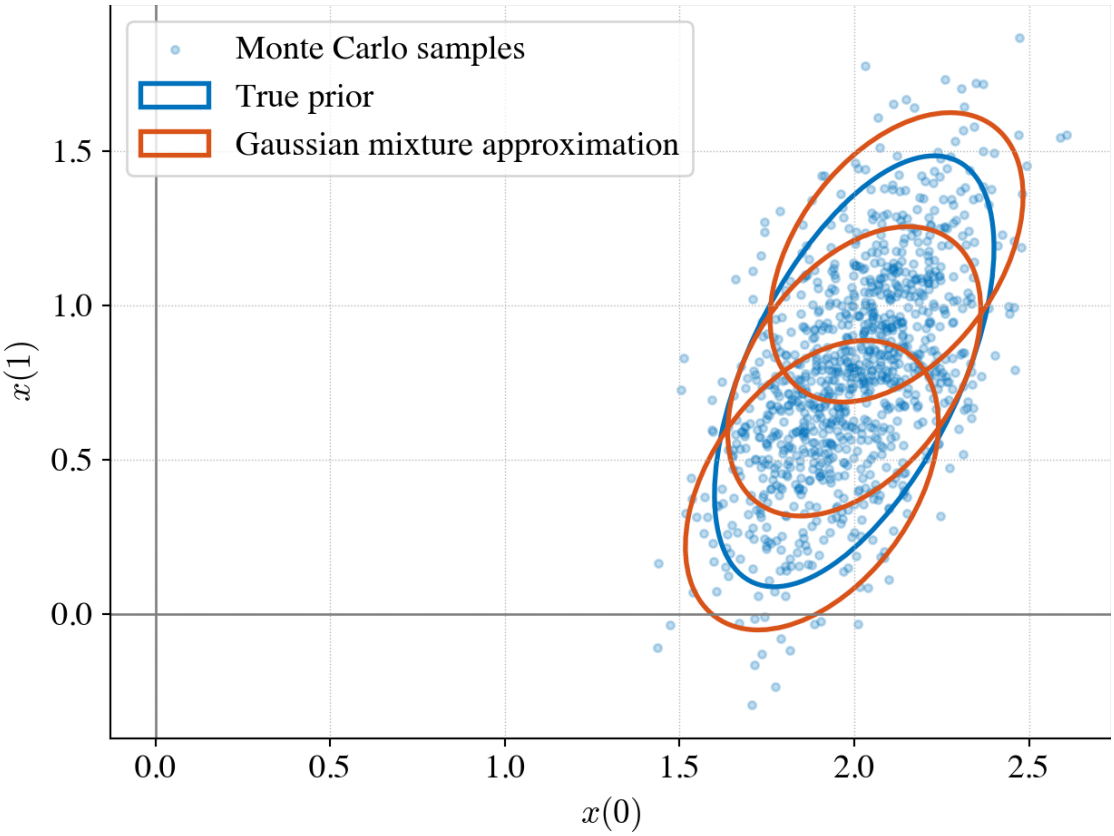

Figure 4.1: Prior density for the Polar-to-Cartesian example. The blue dots represent one thousand of the Monte Carlo samples. The blue ellipse represents the two-sigma boundary of the true prior density. The set of three red ellipses represent the Gaussian mixture approximating the true prior as the prior Gaussian component is split along DoMV.

The Monte Carlo samples and the Gaussian mixture components following the nonlinear update, which now represent the propagated density, are shown in Fig. 4.2 and Fig. 4.6 for the two examples. The Monte Carlo samples help visualize

the true propagated density. Both of these figures illustrate the benefit of splitting and using a Gaussian mixture over a Gaussian representation, as it is evident that any single Gaussian density would be a poor representation of the true propagated density.

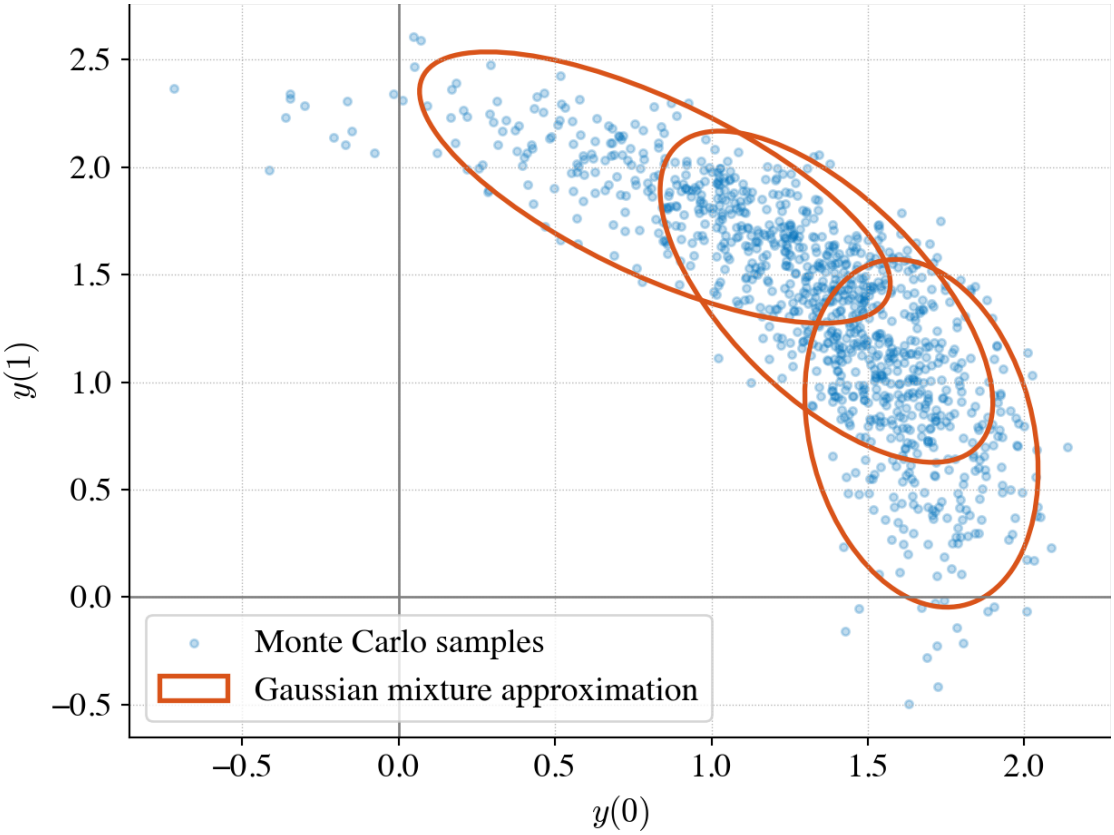

Figure 4.2: Propagated density estimation for the Polar-to-Cartesian example. The blue dots represent one thousand of the Monte Carlo samples after they have been propagated through the nonlinear function. The set of three red ellipses represent the Gaussian mixture approximating the propagated density after split along DoMV.

Fig. 4.3 and Fig. 4.7 show the KL divergence for various directions of split. Firstly, they confirm that the choice of split direction does influence the performance of the Gaussian mixture estimate. They also show that splitting along DoMV results in a better approximation of the true propagated density than splitting along DoNL. In fact, based on the grid search over all possible angles of split, the DoMV is very close to the best angle of split in both of the examples. Note that while the grid search may provide a slightly better angle of split than DoMV, it requires much more computation and is not a feasible option for state vectors of higher dimension.

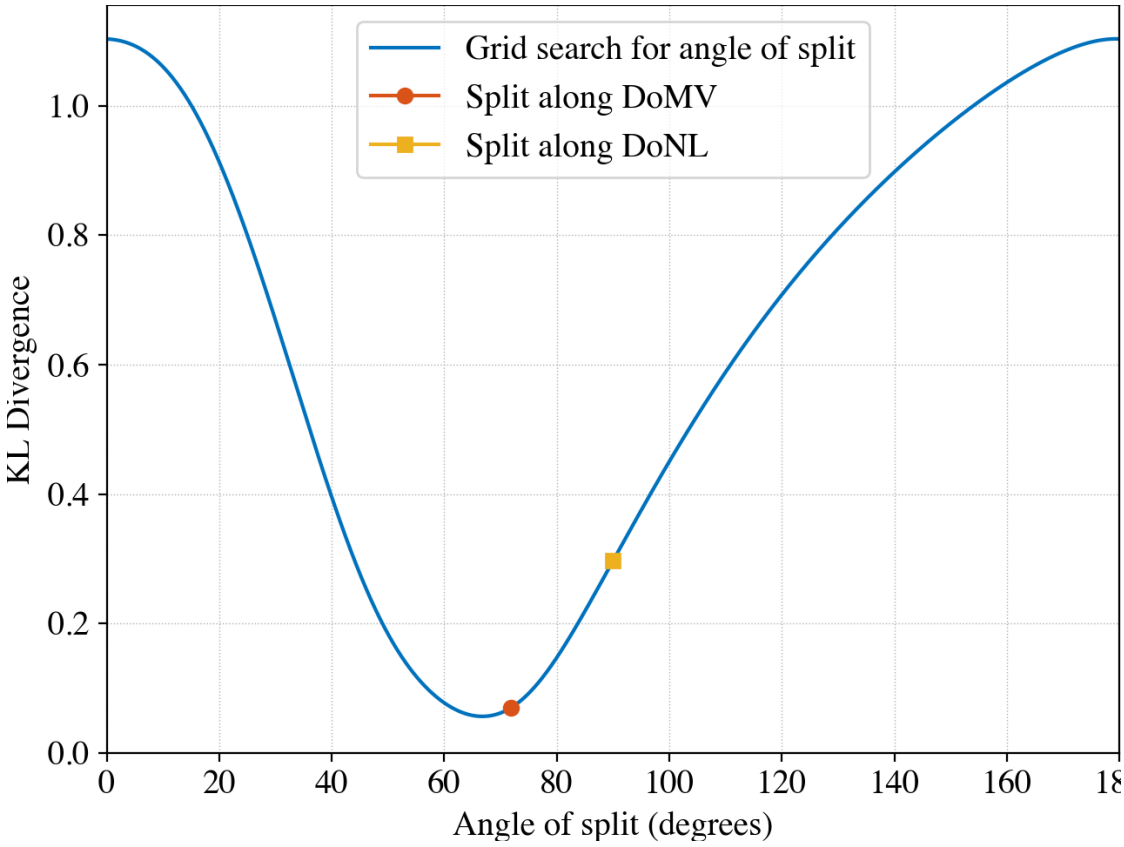

Figure 4.3: KL divergence of the reference for the true propagated density relative to the estimated propagated density as a function of the direction of split for the Polar-to-Cartesian example. The solid blue line shows the KL divergence for various directions of split, and its minimum corresponds to the optimal direction of split. Splitting along DoMV results in closer performance to optimal than the DoNL does. The optimal direction of split, DoMV and DoNL are 66.8°, 71.8° and 90° respectively.

Similarly, Fig. 4.4 and Fig. 4.8 show the ISE distance for various directions of

split. The results are very similar to the KL divergence measure. One notable difference is that the optimal angle of split for ISE distance is closer to DoMV than the KL divergence was. This may be due to the Gaussian splitting library being optimized for ISE distance as detailed in Section 2.1.1.

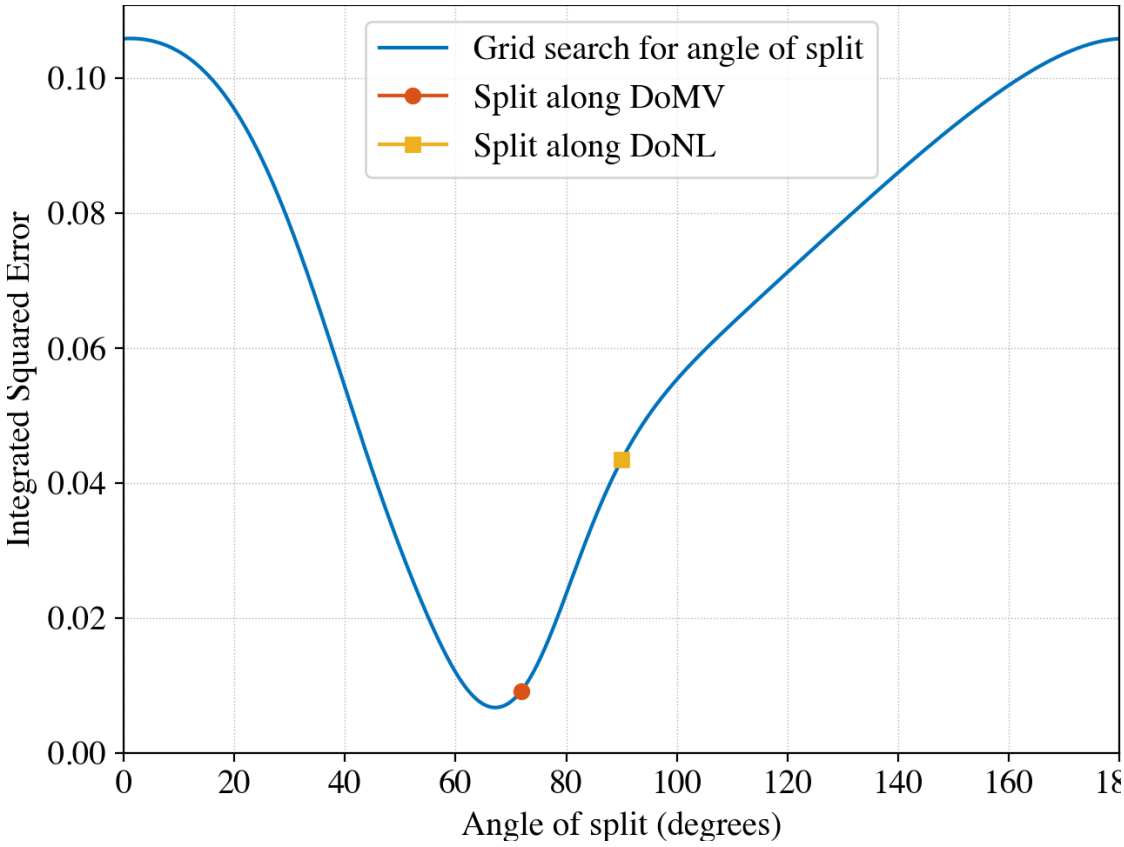

Figure 4.4: ISE distance of the reference of the true propagated density relative to the estimated propagated density as a function of the direction of split for the Polar-to-Cartesian example. The solid blue line shows the ISE distance for various directions of split, and its minimum corresponds to the optimal direction of split. Splitting along DoMV results in closer performance to optimal than the DoNL does. The optimal direction of split, DoMV and DoNL are 67.2°, 71.8° and 90° respectively.

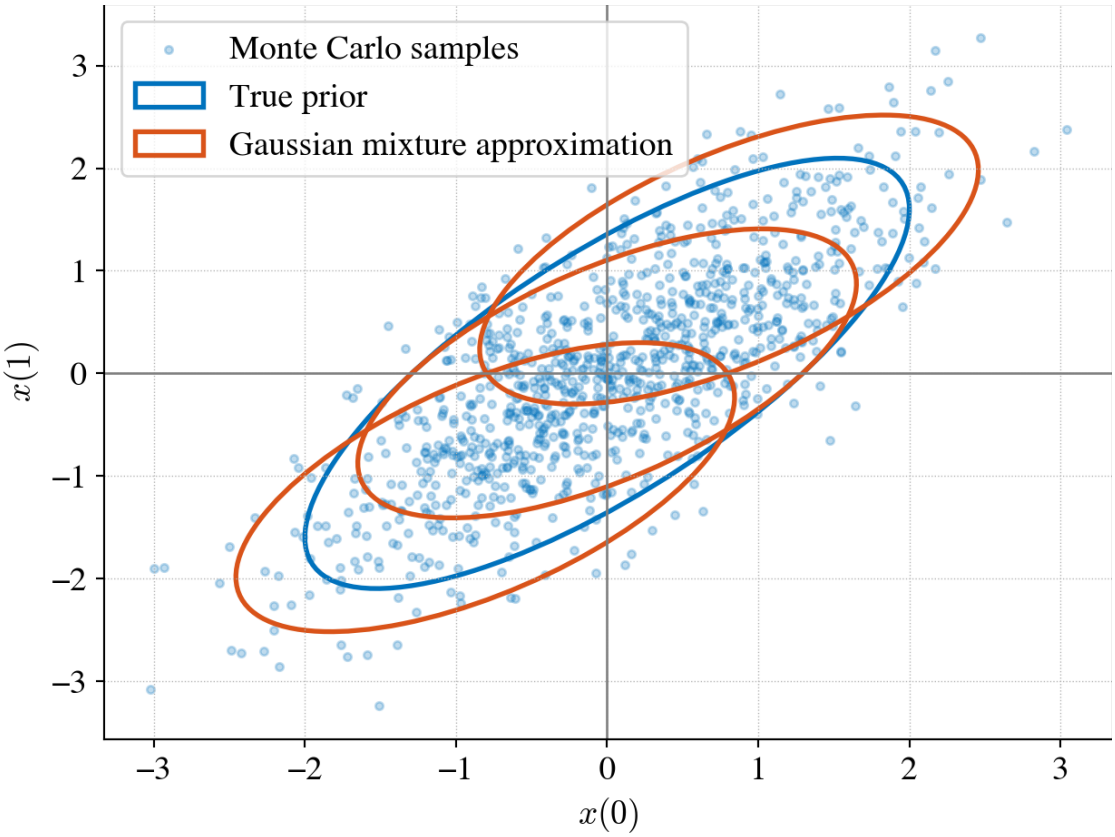

Figure 4.5: Prior density for the Arctan example. The blue dots represent one thousand of the Monte Carlo samples. The blue ellipse represents the two-sigma boundary of the true prior density. The set of three red ellipses represent the Gaussian mixture approximating the true prior as the prior Gaussian component is split along DoMV.

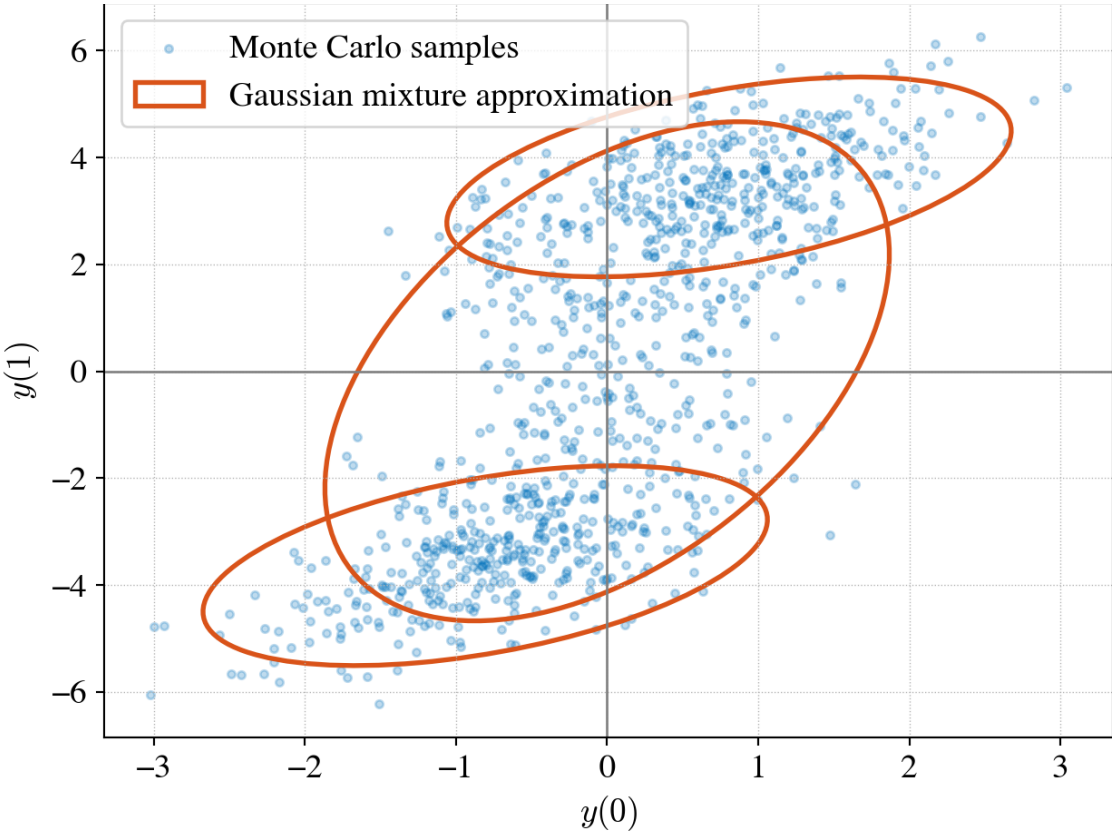

Figure 4.6: Propagated density estimation for the Arctan example. The blue dots represent one thousand of the Monte Carlo samples after they have been propagated through the nonlinear function. The set of three red ellipses represent the Gaussian mixture approximating the propagated density after split along DoMV.

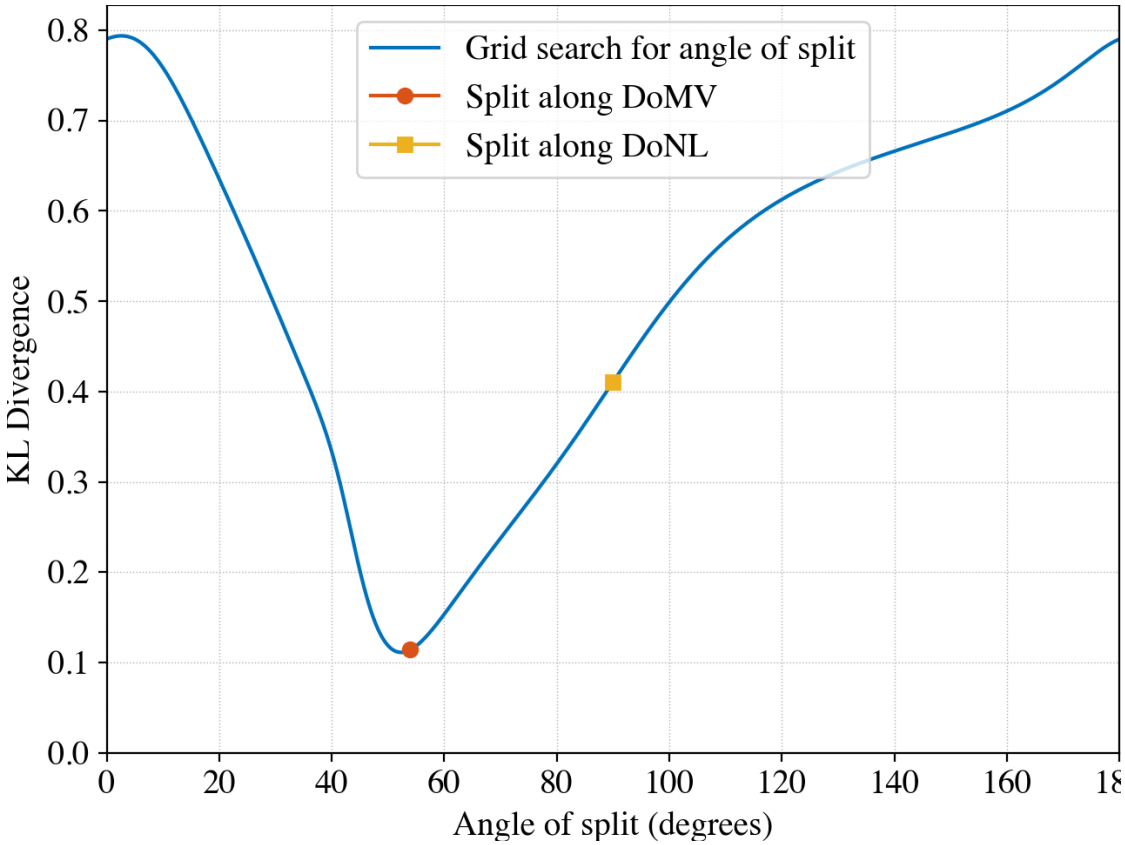

Figure 4.7: KL divergence of the reference for the true propagated density relative to the estimated propagated density as a function of the direction of split for the Arctan example. The solid blue line shows the KL divergence for various directions of split, and its minimum corresponds to the optimal direction of split. Splitting along DoMV results in closer performance to optimal than the DoNL does. The optimal direction of split, DoMV and DoNL are 52.4°, 54.0° and 90° respectively.

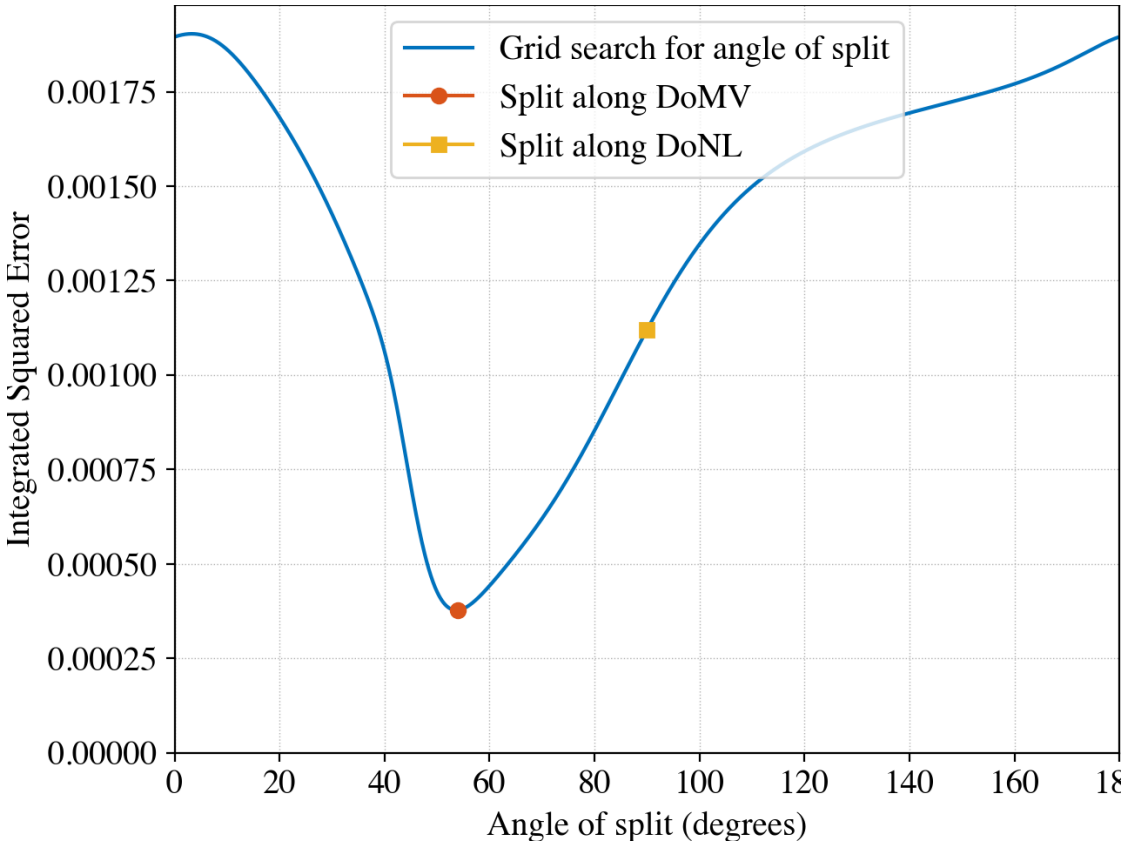

Figure 4.8: ISE distance of the reference for the true propagated density relative to the estimated propagated density as a function of the direction of split for the Arctan example. The solid blue line shows the ISE distance for various directions of split, and its minimum corresponds to the optimal direction of split. Splitting along DoMV results in closer performance to optimal than the DoNL does. The optimal direction of split, DoMV and DoNL are 53.7°, 54.0° and 90° respectively.

5 Application in State Space Tracking

5.1 Brief Overview of State Space Tracking

State space tracking is a Bayesian approach to estimate the state of a dynamic system with a set of noisy observations related to the state. It models the state changes and observations as a sequential process. This is illustrated in Fig. 5.1. The state may change and a new set of observations may be made at each time step. This sequential formulation simplifies the state space tracking problem to that of tracking the state change at a single time step as the same method can be used recursively to also track the state indefinitely.

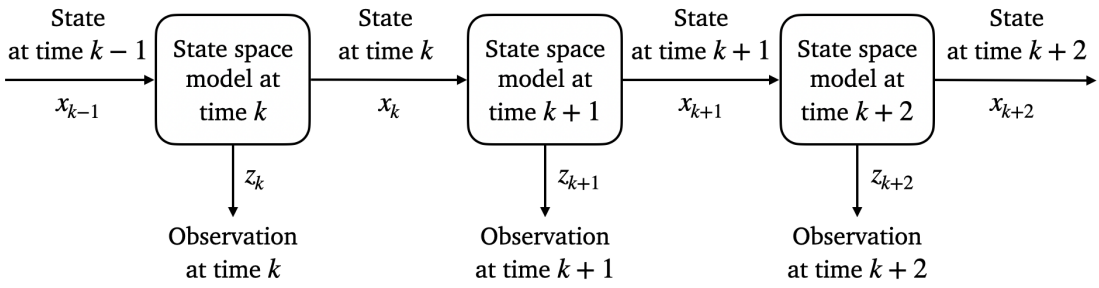

Figure 5.1: Sequential model of a dynamic system for state space tracking.

5.1.1 Notation in This Chapter

This chapter uses different set of symbols and notations than the preceding text. Previously, x and y have been used to respectively represent the input and output of the state transition model, as in (1.1). The same does not work well for state space tracking where the state changes sequentially multiple times. Hence, this chapter uses a different set of symbols and notations as listed in Table 5.1.

Table 5.1: State space tracking symbols and notations.

Symbol	**Description**
x_k	State after the kth time step.
z_k	Observation at the kth time step.
f_k	State transition function at time step k
h_k	Measurement function at time step k
w_k	Process noise at time step k
v_k	Measurement noise at time step k
$p(x_0)$	Initial prior density of the state.
$p\left(x_k \vert z_{1:k}\right)$	Posterior density after kth time step.

5.1.2 State Space Models

Fig. 5.2 shows the state space model at a single time step. The state updates according to a *process model*, which consists of a deterministic state transition

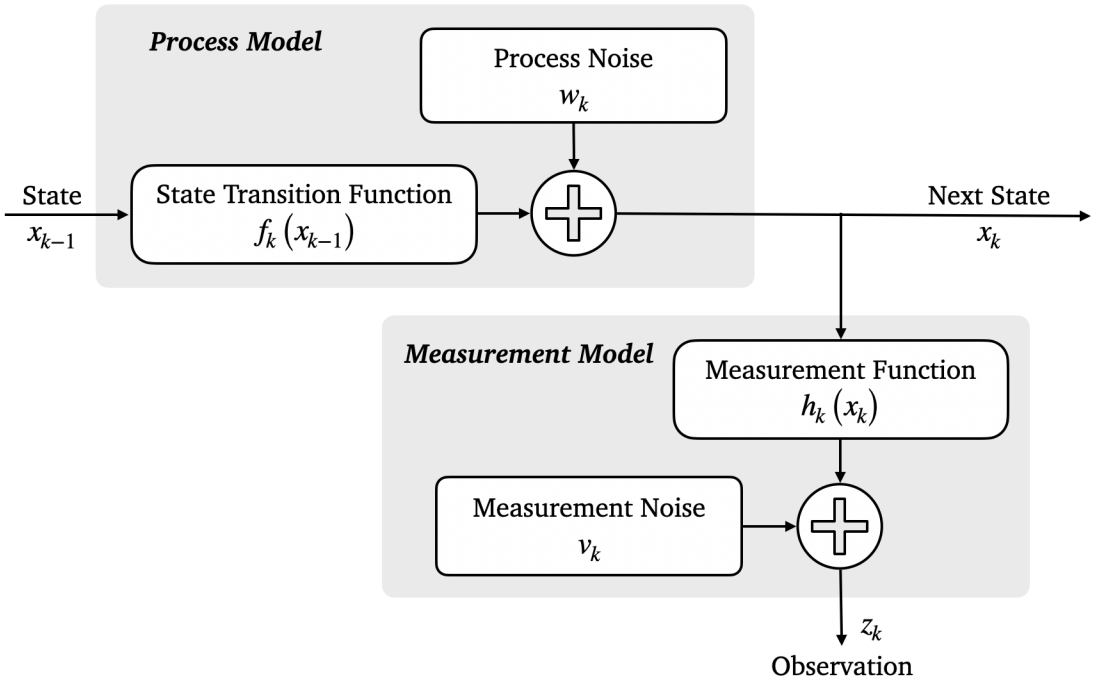

Figure 5.2: Model for the kth time step in state space tracking.

function and a stochastic process noise. Similarly, the observations are modeled by a *measurement model*, which consists of a deterministic measurement function of the state and a stochastic measurement noise.

Using the new notation, the state space model at the kth time step can be represented by the following set of equations.

$$x_k = f_k(x_{k-1}) + w_k \tag{5.1}$$

$$z_k = h_k(x_k) + v_k \tag{5.2}$$

Due to the stochastic nature of the model, the state is a random variable. The

goal of the state space tracking methods is to estimate the posterior density of the state $p(x_k|z_{1:k})$, *i.e.*, the density of the state after time step k, conditional on the set of all observations from the first to the kth time step. Typically this is done recursively, *i.e.*, the posterior density $p(x_k|z_{1:k})$ is estimated from the previous posterior estimate $p(x_{k-1}|z_{1:k-1})$, the model parameters at time step k, and the new observation z_k.

5.2 Relation to Density Propagation

The focus of this work has been on density propagation when a nonlinear function is applied on a random state vector. The same occurs in state space models where nonlinear state transition functions and measurement functions act on the random state vectors. Hence, the method proposed in this thesis, which is to split Gaussian components along DoMV, is also applicable to the broader state space tracking problem when the densities are represented as Gaussian mixtures.

To explore the merit of the proposed method in this broader context, this chapter examines the effect of various directions of split on the tracking performance of a Gaussian mixture filter using a simulated state space tracking example.

5.3 State Space Tracking Example

5.3.1 State Space Model Parameters

The following is an example of a nonlinear state space tracking problem where a simple UKF results in large tracking error. The problem can be thought of as tracking the position of an object, such as a boat, that moves in a two-dimensional plane with constant speed but constantly changing heading. The state is a four-dimensional vector such that

$$x_k = \begin{bmatrix} a_k \\ b_k \\ \theta_k \\ \omega_k \end{bmatrix} \tag{5.3}$$

where a_k and b_k are the Cartesian coordinates of the object's position, θ is the heading angle, and ω is the rate of change of the heading angle. The process and measurement models are nonlinear and time-invariant. The state transition function is

$$f_k\left(x_{k-1}\right) = \begin{bmatrix} a_{k-1} + \gamma\cos(\omega_{k-1}) \\ b_{k-1} + \gamma\sin(\omega_{k-1}) \\ \theta_{k-1} + \omega_{k-1} \\ \omega_{k-1} \end{bmatrix} \tag{5.4}$$

where γ, *i.e.*, the speed of the object, is 2. Note that the output of the function changes nonlinearly only if the input changes along the third component. Thus, the DoNL for the state transition function is $[0\ 0\ 1\ 0]^T$. The process noise is zero-mean Gaussian with covariance Q, *i.e.*,

$$w_k \sim \mathcal{N}(0, Q), \tag{5.5}$$

where

$$Q = \begin{bmatrix} 0.04 & 0 & 0 & 0 \\ 0 & 0.04 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0.0001 \end{bmatrix}. \tag{5.6}$$

The observations are in form of the distance and angle of the object from the origin. So, the measurement function is

$$h_k(x_k) = \begin{bmatrix} \sqrt{a_k^2 + b_k^2} \\ \arctan\left(\dfrac{b_k}{a_k}\right) \end{bmatrix}. \tag{5.7}$$

It is not straightforward to define and estimate the exact direction of maximum nonlinearity for this function. But since the output changes nonlinearly with the first two components of the input, $[1\ 1\ 0\ 0]^T$ is a simplifying and reasonable assumption for DoNL of the measurement function. The measurement noise is zero-mean

Gaussian with covariance R, *i.e.*,

$$v_k \sim \mathcal{N}(0, R), \tag{5.8}$$

where

$$R = \begin{bmatrix} 10 & 0 \\ 0 & 0.1 \end{bmatrix}. \tag{5.9}$$

The initial prior is a Gaussian with mean μ_0 and covariance P_0, *i.e.*,

$$p(x_0) \sim \mathcal{N}(\mu_0, P_0), \tag{5.10}$$

where

$$\mu_0 = \begin{bmatrix} 20 \\ 20 \\ 0 \\ 0 \end{bmatrix} \tag{5.11}$$

and

$$P_0 = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.12}$$

Note that the ω, which is the rate of change of the heading, is a random variable with large initial prior uncertainty. Depending on the value is takes in any given realization, the object might move either clockwise or counterclockwise. This makes the ability to represent multimodal densities important for achieving good tracking performance. Hence, this example is especially challenging for a simple UKF, and suitable for Gaussian mixture filters.

5.3.2 Gaussian Mixture Filtering Algorithm

While UKF and particle filter are well-established and well-defined methods, there is no such consensus on the form the Gaussian mixture filters yet. Fig. 5.3 shows the main steps of the Gaussian mixture filter used in this simulation.

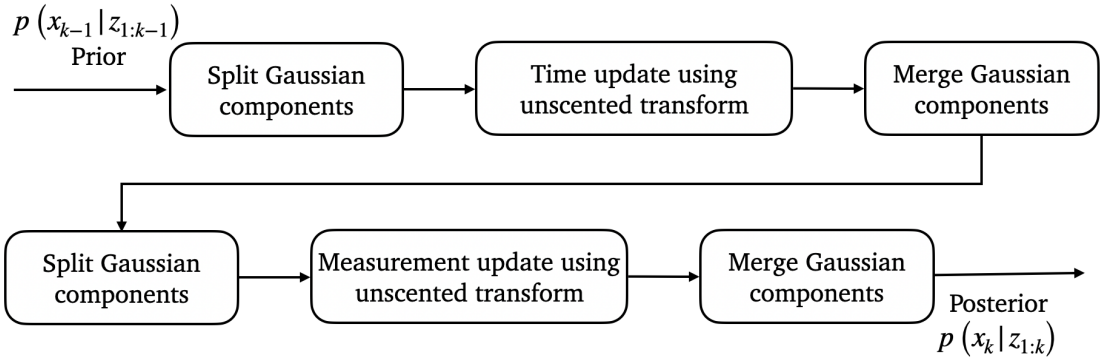

Figure 5.3: Steps in the Gaussian mixture filter.

During the splitting steps, each of the Gaussian components is split into two subcomponents along the chosen direction of split by the method described by Faubel and Klakow in [11]. The time updates and measurement updates are as described by Kotecha and Djuric in Section II-A of [20]. The merge steps are necessary to keep the number of Gaussian components from growing geometrically. The process merges the best pair of Gaussian components, as suggested by Faubel *et al.* in [12], repeatedly until eight of fewer Gaussian components remain.

5.3.3 Results

The simulation includes 100 independent realizations of the process model, with each realization being 50 samples long. The average mean squared error (MSE) of the estimated position acts as the measure of performance. Table 5.2 lists the average MSE for various tracking methods.

Table 5.2: Average MSE for variuous state space tracking methods.

Tracking Method	**Average MSE**
UKF	42.81
GM UKF with split along **DoNL**	25.02
GM UKF with split along **DoPA**	15.46
GM UKF with split along **DoMV**	6.33

Splitting along DoMV results in lower average MSE than splitting along the direction of principal axis (DoPA) or along the DoNL. Note that this finding is very similar to the results for KL divergence or ISE metric for a single time step in Chapter 4.

6 Conclusions

6.1 Contributions

This work reveals and advocates a new direction of split, termed DoMV, for splitting the Gaussian components of a Gaussian mixture. At present, it is common practice to split a Gaussian component along either its principal axis or the direction of nonlinearity. While the former is only a function of the state density, the later is only a function of the system dynamics. This work suggests that when the direction of function nonlinearity (DoNL) is known or can be estimated, the Gaussian components should be split along DoMV. DoMV depends both on the density and system dynamics. It is obtained simply by multiplying the covariance matrix of the Gaussian component with the DoNL. So the extra cost for splitting along DoMV instead of DoNL is negligible. At the same time, it can result in significant performance improvement because splitting along DoMV minimizes the variance of the subcomponents along the DoNL, and a smaller variance makes the linear approximations used by the unscented transform more localized and therefore more accurate. As expected, the examples demonstrated that the DoMV produced a more accurate estimate of the propagated density $q(y)$ than splitting along the DoNL, as measured by KL divergence. The DoMV also was close to the optimal direction of split determined by a grid search to minimize either the KL divergence or the ISE distance.

Another contribution of this work is a concise and robust formulation for splitting a multivariate Gaussian into subcomponents along any direction. This is shown in (2.15) and (2.16).

The evaluation methodology outlines a scheme that uses measures such as KL divergence and ISE distance to compare the estimated probability density with reference probability density, resulting in a more comprehensive measure of performance than the more widely used metric of mean squared error (MSE) between the true state and the mean of the estimated density.

6.2 Open Problems

This work touches on a narrow aspect of the broader Gaussian mixture state space tracking problem. Among the many directions that it can be expanded in, the following are of particular interest.

6.2.1 DoNL Estimation

The computation of DoMV, as presented here, assumes the knowledge of DoNL. While there has been much work on estimating DoNL, it remains an open and challenging problem. For example, most commonly the search for DoNL is performed only along a small number of directions, typically the eigenvectors of the covariance matrix [8, 15, 33]. Using a wider search for DoNL requires much more computation, which scales exponentially with the dimension of the state vector. A computationally practical and general DoNL estimation scheme is not available

at present. Among the available measures, rigorous evaluation of the trade-off between computation and accuracy, and an understanding of how well they work across a range of example systems are also missing. Either estimating DoMV directly, or further progress on robust and practical ways of finding DoNL will broaden the scope of DoMV.

6.2.2 Splitting Strategy

There are many different proposed splitting libraries. For example, DeMars *et al.* in [6] and Vishwajeet and Singla in [32] propose different splitting methods that respectively result in even and odd number of subcomponents. A common understanding of how to choose the best splitting method for a given problem is lacking.

Furthermore, the splitting libraries are pre-optimized and result in a certain number of subcomponents. They do not depend on the degree of system nonlinearity. However, the goal of splitting is to reduce the subcomponent covariance relative to system nonlinearity. One way to balance the two has been to perform the splitting recursively [6, 33] until the measure of nonlinearity for every subcomponent is smaller than a threshold. But this results in much higher computation. A way of estimating how many subcomponents to split into and which splitting library to use based on analysis of the nonlinearity will increase computational efficiency.

6.2.3 Merging Strategy

Splitting increases the number of Gaussian components in the mixture, which also results in more computation. Hence, it is desirable to merge two or more Gaussian components into one if it can be done without introducing significant error in the probability density representation. The challenge with merging lies in finding the subset of Gaussian components whose mixture can be well approximated by a single Gaussian. There have been many proposals for the merging strategy, some of which have been reviewed by Li *et al.* in [22]. A common approach is to find and merge similar Gaussian components, where the similarity between components may be measured by KL divergence, Wasserstein-distance, integrated squared error or other similar measures [4, 24]. But this is not the best strategy as it is not able to merge two or more Gaussian components that are different from each other but can still sum up to a near-perfect Gaussian. For example, consider the case of the splitting algorithm presented in Chapter 2. A good merge algorithm should be able to merge the split subcomponents back into original subcomponent, which the similarity-based methods are not designed to do. More capable merge strategies have been proposed that are based on the similarity between the mixture of the subcomponents and the merged version of the subcomponents [3]. But these also require more computation. Moreover, in all practical proposals, the candidate subcomponents for merge are identified using pair-wise tests. This is sub-optimal, since in some cases, particularly with the splitting library used in this work, testing subsets of three or more subcomponents may be required to find

good candidates for merging. Overall, much more work is required to understand the trade-off between computation and performance for various merging strategies across different applications.

Bibliography

[1] H.H. Afshari, S.A. Gadsden, and S. Habibi. Gaussian filters for parameter and state estimation: A general review of theory and recent trends. *Signal Processing*, 135:218–238, 2017.

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

[3] Tohid Ardeshiri, Karl Granström, Emre Özkan, and Umut Orguner. Greedy reduction algorithms for mixtures of exponential family. *IEEE Signal Processing Letters*, 22(6):676–680, June 2015.

[4] Akbar Assa and Konstantinos N. Plataniotis. Wasserstein-distance-based gaussian mixture reduction. *IEEE Signal Processing Letters*, 25(10):1465–1469, Oct 2018.

[5] Zhe Chen. Bayesian filtering: From Kalman filters to particle filters, and beyond. *Statistics: A Journal of Theoretical and Applied Statistics*, 182:1–69, Jan 2003.

[6] Kyle J. DeMars, Robert H. Bishop, and Moriba K. Jah. Entropy-based approach for uncertainty propagation of nonlinear dynamical systems. *Journal of Guidance, Control, and Dynamics*, 36(4):1047–1057, 2013.

[7] P.M. Djuric, J.H. Kotecha, Jianqui Zhang, Yufei Huang, T. Ghirmai, M.F. Bugallo, and J. Miguez. Particle filtering. *IEEE Signal Processing Magazine*, 20(5):19–38, 2003.

[8] J. Duník, O. Straka, Mahendra Mallick, and Erik Blasch. Survey of nonlinearity and non-gaussianity measures for state estimation. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1845–1852, July 2016.

[9] Jindřich Duník, Ondřej Straka, Benjamin Noack, Jannik Steinbring, and Uwe D. Hanebeck. Directional splitting of Gaussian density in non-linear random variable transformation. *IET Signal Processing*, 12(9):1073–1081, 2018.

[10] Donald Ebeigbe, Tyrus Berry, Michael M. Norton, Andrew J. Whalen, Dan Simon, Timothy Sauer, and Steven J. Schiff. A generalized unscented transformation for probability distributions, 2021.

[11] Friedrich Faubel and Dietrich Klakow. Further improvement of the adaptive level of detail transform: Splitting in direction of the nonlinearity. *2010 18th European Signal Processing Conference*, pages 850–854, 2010.

[12] Friedrich Faubel, John McDonough, and Dietrich Klakow. The split and merge unscented gaussian mixture filter. *IEEE Signal Processing Letters*, 16(9):786–789, Sep. 2009.

[13] Simon Godsill. Particle filtering: the first 25 years and beyond. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7760–7764, May 2019.

[14] Frank Havlak and Mark Campbell. Discrete and continuous, probabilistic anticipation for autonomous robots in urban environments. *IEEE Transactions on Robotics*, 30(2):461–474, April 2014.

[15] Jindřich Havlík and Ondřej Straka. Measures of nonlinearity and non-gaussianity in orbital uncertainty propagation. In *2019 22th International Conference on Information Fusion (FUSION)*, pages 1–8, July 2019.

[16] Joshua T. Horwood and Aubrey B. Poore. Adaptive gaussian sum filters for space surveillance. *IEEE Transactions on Automatic Control*, 56(8):1777–1790, Aug 2011.

[17] Xiao-Li Hu, Thomas B. Schon, and Lennart Ljung. A basic convergence result for particle filtering. *IEEE Transactions on Signal Processing*, 56(4):1337–1348, April 2008.

[18] Marco F. Huber, Tim Bailey, Hugh F. Durrant-Whyte, and Uwe D. Hanebeck. On entropy approximation for gaussian mixture random vectors. *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 181–188, 2008.

[19] S. Julier, J. Uhlmann, and H.F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, March 2000.

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

[21] Qiang Li, Ranyang Li, Kaifan Ji, and Wei Dai. Kalman filter and its application. In *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, pages 74–77, 2015.

[22] Tiancheng Li, Jinya su, Wei Liu, and Juan Corchado Rodríguez. Approximate gaussian conjugacy: Parametric recursive filtering under nonlinearity, multimodality, uncertainty, and constraint, and beyond. *Frontiers of Information Technology & Electronic Engineering*, 18:1913–1939, 09 2017.

[23] Yu Liu, Kai Dong, Haipeng Wang, Jun Liu, You He, and Lina Pan. Adaptive gaussian sum squared-root cubature kalman filter with split-merge scheme for state estimation. *Chinese Journal of Aeronautics*, 27(5):1242–1250, 2014.

[24] D. Raihan and S. Chakravorty. Particle gaussian mixture (pgm) filters. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1369–1376, July 2016.

[25] Matti Raitoharju and Simo Ali-Loytty. An adaptive derivative free method for bayesian posterior approximation. *IEEE Signal Processing Letters*, 19(2):87–90, Feb 2012.

[26] Matti Raitoharju, Simo Ali-Löytty, and Robert Piché. Binomial Gaussian mixture filter. *EURASIP Journal on Advances in Signal Processing*, 2015(1):36, 2015.

[27] Saikat Saha, Pranab Kumar Mandal, Arunabha Bagchi, Yvo Boers, and Johannes N. Driessen. Particle based smoothed marginal map estimation for general state space models. *IEEE Transactions on Signal Processing*, 61(2):264–273, Jan 2013.

[28] H. W. Sorenson and D. L. Alspach. Recursive Bayesian estimation using Gaussian sums. *Automatica*, 7(4):465–479, July 1971.

[29] Ondřej Straka, Jindřich Duník, and Ivo Punčochář. Directional splitting for structure adaptation of bayesian filters. In *2016 American Control Conference (ACC)*, pages 2705–2710, July 2016.

[30] Gabriel Terejanu. An adaptive split-merge scheme for uncertainty propagation using gaussian mixture models. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, pages 1–10, 2011.

[31] R. Van der Merwe and E.A. Wan. The square-root unscented kalman filter for state and parameter-estimation. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 6, pages 3461–3464 vol.6, May 2001.

[32] Kumar Vishwajeet and Puneet Singla. Adaptive splitting technique for gaussian mixture models to solve kolmogorov equation. In *2014 American Control Conference*, pages 5186–5191, June 2014.

[33] Kumar Vishwajeet and Puneet Singla. Adaptive split/merge-based gaussian mixture model approach for uncertainty propagation. *Journal of Guidance, Control, and Dynamics*, 41(3):603–617, 2018.

[34] Caglar Yardim, Zoi-Heleni Michalopoulou, and Peter Gerstoft. An overview of sequential bayesian filtering in ocean acoustics. *IEEE Journal of Oceanic Engineering*, 36(1):71–89, Jan 2011.

[35] Zhihua Zhang, Chibiao Chen, Jian Sun, and Kap Luk Chan. Em algorithms for gaussian mixtures with split-and-merge operation. *Pattern Recognition*, 36(9):1973–1983, 2003. Kernel and Subspace Methods for Computer Vision.