

Dirac Mixture Trees for Fast Suboptimal Multi-Dimensional Density Approximation

Vesa Klumpp and Uwe D. Hanebeck

Abstract—We consider the problem of approximating an arbitrary multi-dimensional probability density function by means of a Dirac mixture density. Instead of an optimal solution based on minimizing a global distance measure between the true density and its approximation, a fast suboptimal anytime procedure is proposed, which is based on sequentially partitioning the state space and component placement by local optimization. The proposed procedure adaptively covers the entire state space with a gradually increasing resolution. It can be efficiently implemented by means of a pre-allocated tree structure in a straightforward manner. The resulting computational complexity is linear in the number of components and linear in the number of dimensions. This allows a large number of components to be handled, which is especially useful in high-dimensional state spaces.

I. INTRODUCTION

Notation

$\underline{x}, \mathbf{x}$	State vector, random vector $\in \mathbb{R}^N$
$f(\underline{x})$	Density function
$F(\underline{x})$	(Cumulative) distribution function
$\delta(\underline{x})$	Multidimensional Dirac delta function
$\delta(x)$	Scalar Dirac delta function
$H(\underline{x})$	Multidimensional Heaviside step function
$H(x)$	Scalar Heaviside step function
$N(\cdot, m, \sigma)$	Scalar Gaussian density with mean m and standard deviation σ
$N(\cdot, \underline{m}, \mathbf{C})$	Multidimensional Gaussian density with mean vector \underline{m} and covariance matrix \mathbf{C}

A. Motivation

Nonlinear processing of random quantities described by (prior) probability density functions in general requires the approximation of the resulting (posterior) densities. Calculation of the true densities is either too complex or even impossible or the resulting densities are simply of a type inconvenient for further treatment. Hence, the true posterior density, i.e., the result of the processing step that might not be explicitly available, is approximated by a density tractable in subsequent processing steps. Many types of generic analytic density representations are available for that purpose, including Gaussian mixtures [1], Edgeworth series expansions [2], and exponential densities [3].

As an alternative representation, we propose to use mixtures of Dirac delta functions (impulse functions), so called Dirac mixtures, for approximating the underlying true densities in analytic form. This is different from representing densities

by means of random samples [4], which is used by the popular particle filters [5], where the appropriate density parameters, i.e., weights and locations of the particles, are typically calculated by means of Monte Carlo techniques [6], [7].

The proposed approach for calculating optimal Dirac mixture approximations for a given density relies on the systematic minimization of a certain distance measure between the two densities. As standard distance measures operating in the density domain such as the Kullback–Leibler divergence [8], its symmetrized version [9] or integral quadratic distances are obviously not well suited for Dirac mixtures, comparison is performed in the distribution domain. As a result, the fundamental property of the Dirac delta function in the density domain can be exploited to simplify processing while the continuity of the corresponding staircase function in the distribution domain is used for comparison purposes.

Several approaches that simultaneously calculate the locations and weights of all components, so-called batch approaches, have already been published [10], [11]. The case of components with equal weights is treated in [10]. A method for the calculation of optimal weights and locations is given in [11]. For both cases, efficient solution procedures for arbitrary true densities based on homotopy continuation have been proposed.

Typically, the number of components required for achieving a certain approximation quality is not known a priori and must be determined as well. Two approaches for doing so are possible on the basis of the available solutions. The first approach relies on trying different numbers of components suitable for the given approximation problem and employs certain heuristics in order to avoid an exhaustive search. For every number of components investigated, a full recalculation is performed [12]. The second approach adds new components one at a time and adjusts all components to guarantee optimality of the resulting approximation [13]. In both approaches, the complexity of the parameter adjustment grows quadratically with the number of components.

The greedy approach proposed in this paper also builds up the final approximate density by adding one component at a time. However, the components are inserted locally without affecting the set of components already placed. This procedure yields suboptimal results at a much lower computational load compared to the optimal approaches, as the complexity is linear in the number of components. Computational savings are especially severe, when an exhaustive search is used for finding the minimum number of components required to satisfy a certain predefined quality measure.

V. Klumpp and U. D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute of Computer Science and Engineering, Universität Karlsruhe (TH), Germany
klumpp@ira.uka.de, uwe.hanebeck@ieee.org

Potential applications of the greedy approximation include the calculation of the density of a function of random variables, long-term prediction [13], state estimation of nonlinear stochastic systems [14], reachability analysis, numerical integration, and even the generation of pseudo-random numbers.

The paper is structured as follows. The following parts of this section give a problem formulation, an overview of prior work, and the key ideas of sequentially approximating arbitrary density functions by means of Dirac mixtures. The approximation is described in detail in Section III. It is shown how a single Dirac delta function can be used to approximate a region of a density function. Furthermore, a procedure for inserting one Dirac component at a time into the approximation is presented. The problem of determining the next region, whose approximation will be refined, is addressed in Section IV. Simulation results showing approximation results in state space, comparison of higher moments between the proposed Dirac positioning and Monte Carlo sampling, and runtimes of the algorithm are shown in Section V. The paper concludes with a discussion of the proposed new approximation approaches and an outlook to future work.

B. Problem Formulation

We consider a continuous-valued random vector $\tilde{\underline{x}} \in \mathbb{R}^N$ that is approximated by a discrete random vector \underline{x} taking on a finite set of values from the set $\mathcal{D} = \{\underline{x}_1, \underline{x}_2, \dots, \underline{x}_L\}$ with $|\mathcal{D}| = L$. The given random vector $\tilde{\underline{x}}$ is characterized by a true density function $\tilde{f}(\underline{x})$.

The goal is to calculate appropriate locations \underline{x}_i , $i = 1, 2, \dots, L$ and corresponding probabilities w_i , $i = 1, 2, \dots, L$ in such a way that the discrete approximation is in some way close to the true random vector $\tilde{\underline{x}}$.

C. Prior Work

A very common and popular method of approximating a probability distribution function is given by random sampling [15]. However, this method suffers from poor convergence as the samples are drawn independently from the underlying distribution. Furthermore, the approximation produces random results and in addition, depends on the sampling algorithm used.

An alternative to random sampling is to use quasi-random sample sequences that are generated in a deterministic way, which ensures reproducible results. Unfortunately, these Quasi-Monte Carlo techniques typically allow a sampling of uniform densities only. Much work has been done on the field of low-discrepancy sequences that try to minimize the discrepancy, a maximum deviation of a uniform distribution over a unit hyper-cube and its sample representation. Several of these sequences, like Latin Hypercube Sampling [16] or low-discrepancy sequences [17] exist, unfortunately a minimum discrepancy bound, and therefore the best possible sequence, is not known [18], [19]. The generalization of these approaches to the sampling from nonuniform distributions is not simple as optimality is not preserved by a mapping from uniform to arbitrary distributions.

D. Key Ideas of the Paper

The sequential approximation procedure that will be introduced in the following performs the desired approximation by iteratively partitioning the state space, where each partition is represented by a Dirac delta component. It is specifically designed in such a way that it can be applied to high-dimensional problems, which is achieved by three main ingredients.

First, a given region is partitioned into just two subregions independent of the number of dimensions involved. Since every region corresponds to a hyper-rectangle, splitting is performed along the axes of the coordinate system. Hence, there are N possible partitions for a given region, where N is the number of dimensions. As a result, with every partitioning step, the number of Dirac delta components just increases by one.

Second, the quality of the approximation is calculated by means of a distance measure. According to a global distance measure, all components of the approximation density would be dependent and could not be calculated sequentially without affecting components already placed. In order to simplify the approximation problem, the individual regions are treated separately, i.e., a distance measure is minimized for every region. A global distance measure is just used for assessing the total approximation quality.

Third, the complexity of calculating the optimal Dirac delta representation for a given region would typically still increase exponentially with the number of dimensions N . Hence, the marginals of the true density are used for placing the Dirac delta components. Of course, the global marginals are not useful as the true density typically is not composed of a product of its marginals. Here, we define submarginals, i.e., the marginals corresponding to the true density within an individual region of the state space. As a result, the complexity of calculating the Dirac delta representation for a given region increases linearly with the number of dimensions N .

II. DIRAC MIXTURES

The discrete random vector \underline{x} is characterized by a Dirac mixture density of the form

$$f(\underline{x}, \underline{\eta}) = \sum_{i=1}^L w_i \delta(\underline{x} - \underline{x}_i) , \quad (1)$$

where the components $\delta(\underline{x} - \underline{x}_i)$ are multidimensional Dirac delta functions defined by

$$\delta(\underline{x} - \underline{x}_i) = \prod_{j=1}^N \delta(x^{(j)} - x_i^{(j)}) .$$

The scalar Dirac delta functions are defined by

$$\delta(x - x_j) = \begin{cases} \text{undefined,} & x = x_j \\ 0, & \text{elsewhere} \end{cases}$$

with

$$\int_{-\infty}^{\infty} \delta(x - x_j) dx = \int_{x_j - \epsilon}^{x_j + \epsilon} \delta(x - x_j) dx = 1$$

for some $\epsilon > 0$. w_i are weighting factors with $w_i > 0$ and $\sum_{i=1}^N w_i = 1$.

The parameter vector $\underline{\eta}$ contains both the weighting factors and the locations of the individual Dirac functions according to

$$\underline{\eta} = [w_1, w_2, \dots, w_L, x_1, x_2, \dots, x_L]^T .$$

Our goal is to approximate the true density $\tilde{f}(\underline{x})$ by a Dirac mixture density $f(\underline{x})$ in such a way that a certain distance measure G between the two densities is below a prespecified threshold. As distance measures in the density domain are typically not well defined for Dirac mixture densities, the first key idea is to employ distance measures between (cumulative) distributions.

The distribution function corresponding to the true density $\tilde{f}(\underline{x})$ is denoted by $\tilde{F}(\underline{x})$. The distribution function corresponding to the approximate density $f(\underline{x}, \underline{\eta})$ is denoted by $F(\underline{x}, \underline{\eta})$ and is given by a Heaviside mixture according to

$$F(\underline{x}, \underline{\eta}) = \sum_{i=1}^L w_i H(\underline{x} - \underline{x}_i) .$$

$H(\underline{x} - \underline{x}_i)$, $i = 1, \dots, L$ are multidimensional Heaviside step functions defined by

$$H(\underline{x} - \underline{x}_i) = \prod_{j=1}^N H(x(j) - x_i(j)) ,$$

where the scalar Heaviside step functions are given by

$$H(x - x_i) = \begin{cases} 0, & x < x_i \\ \frac{1}{2}, & x = x_i \\ 1, & x > x_i \end{cases} .$$

In the following, we restrict our focus to quadratic distance measures between the true distribution $\tilde{F}(\underline{x})$ and its Dirac mixture approximation $F(\underline{x}, \underline{\eta})$.

Hence, the second key idea pursued in this paper is a greedy type sequential approximation of the true density $\tilde{f}(\underline{x})$. Instead of optimizing the approximation by means of minimizing a global distance measure, an approach is utilized, which improves the approximation in a greedy way, i.e., the action is taken that maximizes the improvement of the approximation in one optimization iteration. This typically does not lead to an optimal approximation, but it has the advantage of being very fast and easy to implement.

Another important fact is that the approximation of the true density $\tilde{f}(\underline{x})$ is not homogeneous over the state space. Regions of high interest, i.e., regions with a high probability mass, are approximated more precisely than regions, which have low probability. The approximation procedure incorporates an adaptive refinement of the regions of interest.

In the optimal multi-dimensional approximation of density functions with Dirac mixtures, the complexity grows exponentially with the number of dimensions. This problem can be solved by working on the individual marginals of the regions of high interest, which leads to a linear growth of algorithmic complexity only, and thus, dramatically reduces

computational effort by maintaining a high quality of the approximation result.

III. APPROXIMATION APPROACH

A. Optimizing a Given Region

We consider an axis-aligned hyper-rectangle in \mathbb{R}^N given by a set of lower limits x_i^l , $i = 1, \dots, N$ and a set of upper limits x_i^u , $i = 1, \dots, N$ for characterizing a region in the considered state space.

Example III.1 (Two-dimensional Region) A two-dimensional region is defined by its axis-aligned borders. In every dimension the region can be handled separately and its extend can be given by an interval $[x_i^l, x_i^u]$. So every point $\underline{p} = [p_1, \dots, p_N]^T$ is inside of the region, if $p_i \in [x_i^l, x_i^u]$ holds for every dimension i .

Definition III.1 (Submarginals) The submarginals of the given true density $\tilde{f}(\underline{x})$ in the considered region are given by

$$\tilde{f}(x_i) = \int_{x_1^l}^{x_1^u} \dots \int_{x_{i-1}^l}^{x_{i-1}^u} \int_{x_{i+1}^l}^{x_{i+1}^u} \dots \int_{x_N^l}^{x_N^u} \tilde{f}(\underline{x}) dx_N \dots dx_{i+1} dx_{i-1} \dots dx_1$$

for $i = 1, \dots, N$. The respective cumulative probability distribution $\tilde{F}(x_i)$

$$\tilde{F}(x_i) = \int_{x_i^l}^{x_i^u} \tilde{f}(x_i') dx_i'$$

is given by integrating the corresponding submarginal.

Example III.2 (Two-dimensional Submarginals) In the case of two-dimensional densities, the submarginals $\tilde{f}(x_i)$ for dimension i are given by integrating the true density function \tilde{f} over the dimension $3 - i$ with

$$\tilde{f}(x_1) = \int_{x_2^l}^{x_2^u} \tilde{f}(x_1, x_2) dx_2, \quad x_1 \in [x_1^l, x_1^u], \quad \text{and}$$

$$\tilde{f}(x_2) = \int_{x_1^l}^{x_1^u} \tilde{f}(x_1, x_2) dx_1, \quad x_2 \in [x_2^l, x_2^u] .$$

Note that the submarginals are only defined on the corresponding interval $[x_i^l, x_i^u]$.

For many interesting densities, the submarginals are either directly given or can be calculated analytically.

Example III.3 (Submarginals of Gaussian Density) The submarginals of a Gaussian density can easily be written by means of the error function. By assuming an axis-aligned Gaussian density

$$\tilde{f}(x_1, x_2) = N \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \right)$$

$$= N(x_1, m_1, \sigma_1^2) \cdot N(x_2, m_2, \sigma_2^2)$$

the submarginal for dimension 1 is simply given as

$$\tilde{f}(x_1) = \frac{1}{2} \left(\operatorname{erf} \left(\frac{x_2^u - m_2}{\sqrt{2} \sigma_2} \right) - \operatorname{erf} \left(\frac{x_2^l - m_2}{\sqrt{2} \sigma_2} \right) \right) N(x_1, m_1, \sigma_1^2).$$

The submarginal for the other dimension is calculated analogously.

Lemma III.1 (Weight of Dirac Component) The weight of the Dirac component representing the considered region is given by the probability mass of the true density $\tilde{f}(\underline{x})$

$$w = \int_{x_1^L}^{x_1^U} \cdots \int_{x_N^L}^{x_N^U} \tilde{f}(\underline{x}) dx_N \dots dx_1$$

inside the region.

Based on the submarginals for the considered region, an appropriate but suboptimal location of the Dirac component is calculated by treating the individual dimensions separately. As a result, the complexity just increases linearly with the number of dimensions.

Theorem III.1 (Location of Dirac Component) The optimal location \hat{x}_i in dimension i of the Dirac component representing the considered region with respect to the Cramér–von Mises distance [11] is given by the solution of

$$2\tilde{F}(\hat{x}_i) = \tilde{F}(x_i^U) - \tilde{F}(x_i^L)$$

for all dimensions $i = 1, \dots, N$.

Remark III.1 The location \hat{x}_i from Theorem III.1 corresponds to the local median of the submarginal in the considered region. This is equivalent to splitting the total probability mass into two equal submasses.

B. Splitting a Given Region

The splitting line in 2D, the splitting plane in 3D, and the splitting hyperplane in higher dimensions are axis-aligned so that only one dimension is affected. In addition, they always contain the location of the Dirac component representing this region. Hence, there are N possible splittings.

The splitting of a region given by the lower and upper limits $x_i^L, x_i^U, i = 1 \dots N$ in an arbitrary dimension j results in two new regions. The limits of the new regions differ only in the j -th component, because the axis-aligned splitting affects one dimension only. In the following these regions are named *region 1* and *region 2*, independent of the splitting dimension. The limits of a specific region are defined as $x_i^{L,r}, x_i^{U,r}$ with $r = 1, 2$ in order to distinguish them from the original limits.

The resulting components $x_i^{L,1}$ and $x_i^{U,1}$ are given for region 1 as

$$x_i^{L,1} = x_i^L, \quad x_i^{U,1} = \begin{cases} x_i^U, & i \neq j \\ \hat{x}_i, & i = j \end{cases}$$

and for region 2 as

$$x_i^{L,2} = \begin{cases} x_i^L, & i \neq j \\ \hat{x}_i, & i = j \end{cases}, \quad x_i^{U,2} = x_i^U.$$

The cost of evaluating all possible splittings grows linearly with the number of dimensions. The two different splittings in 2D space are shown in Figure 1.

The strategy pursued in this paper is to split the region in the dimension i

$$i = \arg \max_i \{|x_i^U - x_i^L|\} \quad (2)$$

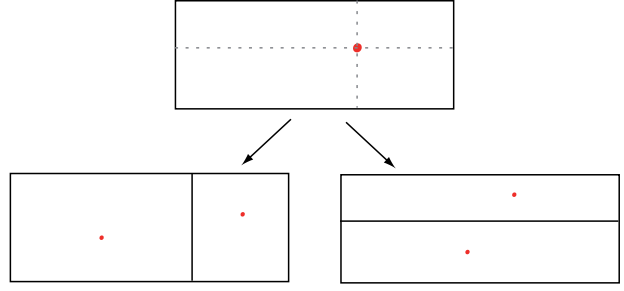


Fig. 1: Visualization of different splittings of a two-dimensional region. The splitting lines halve the probability mass of the original region and are aligned to one of the axis directions.

with maximum width. This avoids the computation and evaluation of the N different splittings and reduces the computational effort. This results in a regular splitting of the complete approximation and thus a more regular Dirac positioning.

Remark III.2 The complexity of selecting the splitting dimension depends linearly on the number of dimensions. Although the effort to compute (2) can be seen as practically constant in comparison to the overall algorithm runtime.

Example III.4 (Region Splitting in 2D) This example shows the splitting result of the Dirac tree approximation of a two-dimensional density \tilde{f} . The true density is given as a Gaussian mixture density

$$\tilde{f} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) = w_1 \cdot N \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \underline{m}_1, \mathbf{C}_1 \right) + w_2 \cdot N \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \underline{m}_2, \mathbf{C}_2 \right)$$

with weights $w_1 = 0.3, w_2 = 0.7$, means

$$\underline{m}_1 = [0.3, 0.6]^T, \quad \underline{m}_2 = [0.6, 0.3]^T,$$

and covariance matrices

$$\mathbf{C}_1 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.05 \end{bmatrix}, \quad \mathbf{C}_2 = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.1 \end{bmatrix}.$$

The result after 15 consecutive splittings is shown in Figure 2.

C. Sequential Partitioning

Sequential partitioning of different regions is needed to refine the approximation. This is achieved by splitting one region in every iteration. First, the algorithm is initialized with a single element. This element contains the region considered for approximation, the Dirac weight, and the Dirac position, derived from the equations in Section III-A. From now on, we consider a list containing the set of all these elements. Then, a selection mechanism is utilized that determines the next splitting region, which is described in detail in Section IV. This region is replaced in the list by its splitting according to Section III-B. This leads to a subsequent minimization of a distance measure between the approximation and the true density. Splitting is repeated until the desired number of Dirac mixture components is reached or the quality of the approximation falls below a given threshold. The resulting

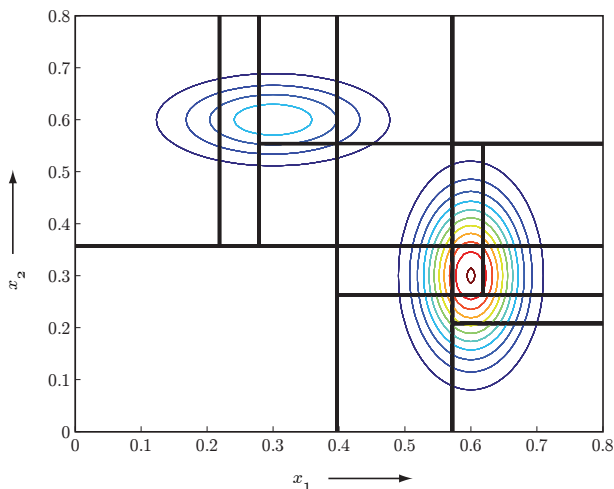


Fig. 2: Splitting of a two-dimensional Gaussian mixture density. The rectangular regions show the regions constructed by the splitting algorithm.

greedy Dirac mixture approximation algorithm is shown in pseudocode notation in Algorithm 1.

Note that with every iteration, the number of Dirac mixture components increases by one, because one element is replaced by two new elements in the list. This can be seen as a tree structure, where the elements of the list are leaf nodes. New elements in the list refine the approximation of the removed one. The corresponding binary tree is shown in Figure 3, where the first element is represented as the root node and the two child nodes represent the resulting elements after the splitting. After 7 sequential splittings, the number of Dirac mixture components increased to 8. In the tree representation these components correspond to the leaf-nodes.

Algorithm 1 Greedy Algorithm for Sequential Dirac mixture approximation

- 1: start with one element in the list
 - 2: **repeat**
 - 3: determine next splitting element (see Algorithm 2) and remove it from list
 - 4: select splitting direction
 - 5: perform split
 - 6: insert the two results into list
 - 7: **until** Maximum number of components obtained (or maximum distance below threshold)
-

IV. REGION SELECTION

In order to keep a good approximation quality over the whole state-space, the selection of the next splitting region is essential. Generally, the region that gives the maximum improvement in approximation quality is chosen. Within the proposed algorithm the improvement depends on the probability mass assigned to the region. Hence, regions with high probability mass are potential splitting candidates.

Because of the anytime functionality of the proposed algorithm, selecting the next splitting candidate is of great importance. That means, after every approximation step the algorithm is able to return and give the best possible coverage for the given number of components. In order to obtain a good approximation quality over the complete state-space, the selection of the splitting region has to fulfill the following properties. First, the selection has to cover the relevant regions as quickly as possible. Second, it has to be symmetric in order to maintain the quality of the first and other odd central moments of the approximation. Here, it is assumed that the locations of the regions in state-space are correlated with their positions in the splitting tree. Hence, the regions are selected based on the following tree structure.

The splitting described in the previous section produces regions with equal probability mass in every layer leading to a tree structure shown in Figure 3. The root node, which represents the first region, holds the complete probability mass, whereas its children nodes, which result from the first splitting, halve the corresponding probability mass. This splitting is applied to all nodes in the tree, resulting in layers with equal probability mass.

The dispersion and symmetry in selecting the splitting region is achieved by an alternating switching at the root node of every subtree recursively. The selection begins at the root node and descends to a leaf, which will be split. In every node, the descent to the left-hand or right-hand side is determined by a simple switch that is toggled everytime the selection algorithm traverses it. This can be described by a virtual state that is assigned to every node and keeps track of the next descending direction. A simple, but important, assumption in this algorithm is the initial state of the tree, which is defined recursively in the following way: the initial state of the right sub-tree is the mirrored (or toggled) state of the left sub-tree. An example initial state configuration is visualized in Figure 3 by the arrows in every non-leaf node.

Instead of storing the state of every node in the complete tree, only n bits for layer n , containing 2^n regions, are needed to compute and address the next splitting region. The algorithm is shown in Algorithm 2. Note that in line 1, i can easily be computed by knowing the actual number of Dirac components, whereas only the lower n bits of i have to be considered. The resulting number is the index that addresses all regions in the considered layer. This procedure is similar to pseudo random number generation [20], [17], especially to the generation of (t, s) -sequences introduced in [18].

Algorithm 2 Select the next splitting region

- 1: $i \leftarrow$ number of Diracs processed in actual layer
 - 2: **for** every bit in i , beginning with MSB **do**
 - 3: **if** bit is set **then**
 - 4: toggle next lower bit
 - 5: **end if**
 - 6: **end for**
 - 7: **return** i
-

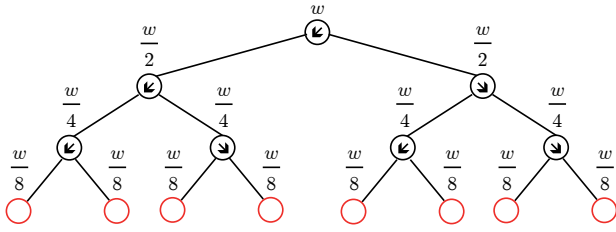


Fig. 3: Subtree of the considered tree structure, where the leaf-nodes correspond to the list of all current regions. The probability w is assigned to the root node and split for every child. A probability is assigned to every region and the probabilities of all leaf-nodes sum up to w .

V. SIMULATION RESULTS

A. Reconstruction of Continuous Densities

The applicability of the proposed approximation approach for multi-dimensional densities is shown by some simple density approximations. First, a simple two-dimensional Gaussian density is approximated by different numbers of Dirac components. The true density \tilde{f} is given as a Gaussian density with zero mean and an identity covariance matrix. Figure 4 shows the approximations for various numbers of Dirac components and the true density. The locations of the Dirac components focus on regions with high probability mass of the true density function. Usually, the Dirac components can have different weights. For simplicity, only approximations whose number of Dirac components is given by powers of 2 are shown here. This leads to the same weights for all Dirac components and thus simplifies visualization. Note that for the symmetric Gaussian density, the Dirac components are placed symmetrically, too.

Figure 5 shows the approximation results for the Gaussian mixture density introduced in Example III.4. It can be seen that the two Gaussian components are approximated with a different number of Dirac components. This is due to the different weights of the Gaussian components and therefore different probability mass distributions over the state-space. The effect of the Dirac components concentrating at locations with high probability mass can be seen, too.

B. Central Moments

In this section, the Dirac approximation is compared to random sampling, as typically used in Monte Carlo techniques, like the well known particle filter. The true density function \tilde{f} , the Gaussian mixture density from example III.4, is approximated by both approaches.

The relative errors of the mean and the 2nd – 4th central moments of the first dimension x_1 are computed for a growing number of components (1–500). In order to smooth the results of the random sampling, the approximation was performed 100 times with independent sample sets. The root mean square errors of the relative moments of both approximations are displayed in Figure 6.

It can be seen clearly, that the relative errors of the Dirac approximation decrease faster in magnitudes than the errors

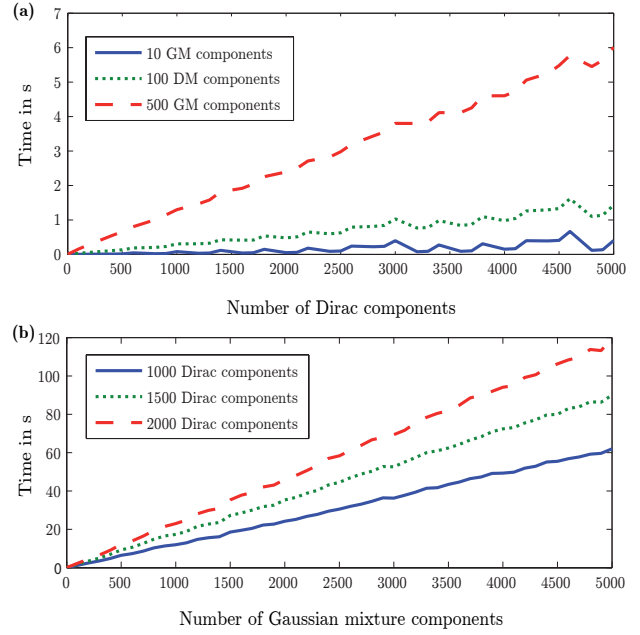


Fig. 7: (a) Computing time against number of dirac components for the approximation of 10 (blue), 100 (green; dotted), and 500 (red; dashed) Gaussian mixture components. (b) Computing time against number of Gaussian mixture components. Approximation with 1000 (blue), 1500 (green; dotted), and 2000 (red; dashed) Dirac components.

of Monte Carlo sampling. Even with a low number of Dirac components, a very good approximation of the complete density (regarding the first moments) is given.

C. Runtime Measurements

For runtime measurements, two-dimensional Gaussian mixture densities were approximated by means of Dirac mixtures. Simulation results show that the runtime is linear in the number of Gaussian components and in the number of Dirac components. In Figure 7 (a), the approximation time is shown depending on the number of Dirac components used for approximation. Figure 7 (b) shows the approximation time depending on the number of Gaussian mixture components that have to be approximated. The runtime measurements were taken in MATLAB[®] calling a C# sub-routine that computes the Dirac approximation.

D. Stochastic Prediction

A two-dimensional discrete time linear system model is given by

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{v}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{v}_k \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha \end{bmatrix} + \begin{bmatrix} \mathbf{w}_x \\ \mathbf{w}_v \end{bmatrix}, \quad (3)$$

with a hysteresis

$$\alpha = \begin{cases} -at, & \mathbf{x}_k > 0.25 \\ at, & \mathbf{x}_k < -0.25 \\ 0, & -0.25 \leq \mathbf{x}_k \leq 0.25 \end{cases},$$

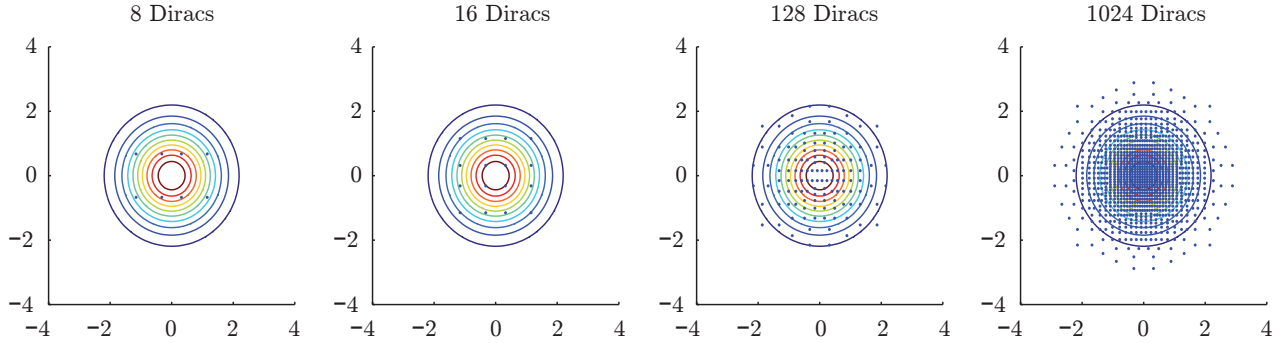


Fig. 4: Approximation result of a two-dimensional Gaussian density with 8, 16, 128, and 1024 Dirac components.

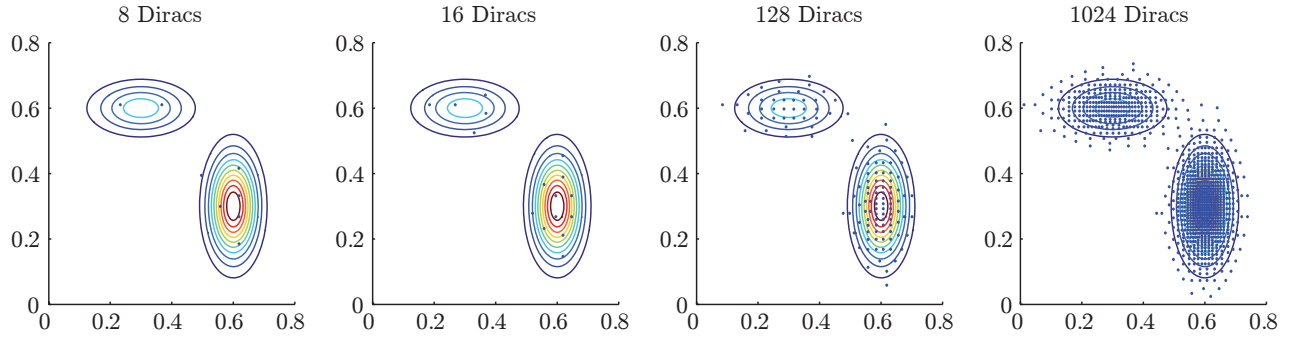


Fig. 5: Approximation result of a two-dimensional Gaussian mixture density with 8, 16, 128, and 1024 Dirac components.

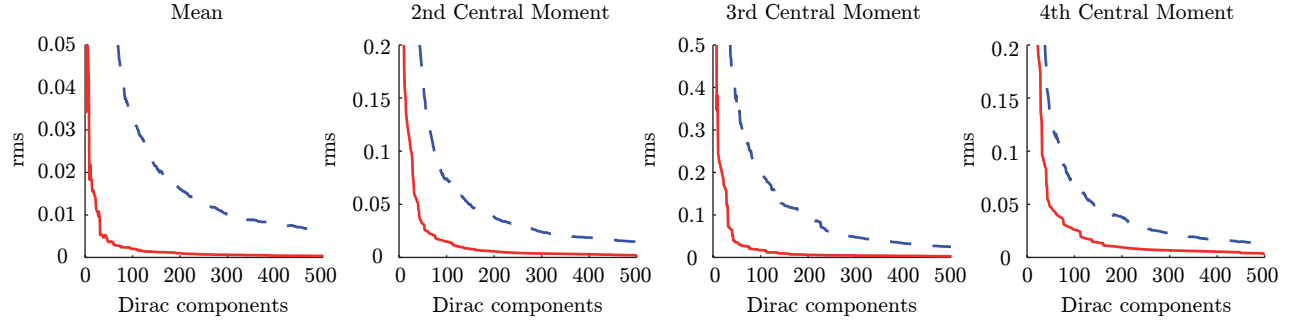


Fig. 6: Root mean square of relative errors of mean and central moments of the density approximations. Random sampling (blue; dashed) and greedy Dirac approximation (red) are shown for a growing number of Diracs/particles used.

which defines a position dependent linear acceleration. The constants for acceleration and time discretization are $a = 0.05$ and $t = 0.05$. The system noise $[\mathbf{w}_x, \mathbf{w}_v]^T$ is given as zero mean Gaussian noise with covariance $\mathbf{C}_w = \text{diag}(0.02, 0.02)$.

Here, the greedy Dirac approximation is used to represent the continuous prior density by means of Dirac components, which are propagated through the system. Due to the system noise in (3) the resulting density is given as a Gaussian mixture density, which is reapproximated for the following processing step.

The Dirac mixture approximation using 100 Dirac components is compared to the particle filter with 4000 particles. The absolute differences d_i of the i -th root of the first i central moments, $i = 1, \dots, 4$, for 50 consecutive prediction steps, is shown in Figure 8. It can be seen clearly, that the

quality of the Dirac approximation is substantially better than the particle approximation, even the number of particles is 40 times higher. The differences of the moments for the particle filter are substantially higher, except for the minimum difference of the moments for the particle filter at time step 18 due to an occasional intersection with the true mean.

VI. DISCUSSION AND FUTURE WORK

This paper introduced a greedy algorithm for calculating Dirac mixtures approximating arbitrary multi-dimensional probability functions. Dirac components are added sequentially one at a time, which allows to minimize a distance measure between the approximation and the true density. It also enables to efficiently determine the required minimum number of components for achieving a certain task. A

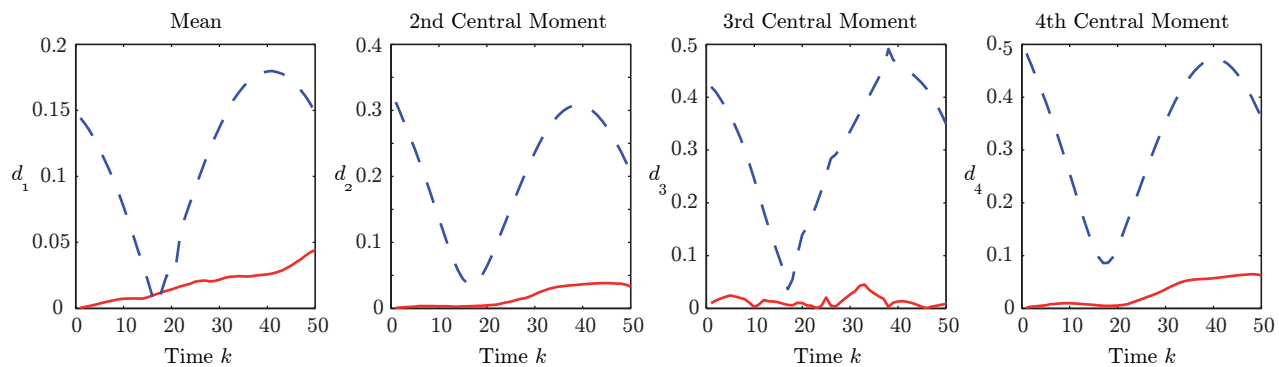


Fig. 8: Visualization of simulation results of Subsection V-D. Absolute differences d_i , $i = 1, \dots, 4$ of mean and 2nd to 4th central moments of substate x for the greedy Dirac approximation (red) and particle filter (blue; dashed) are shown for the first 50 time steps.

non-homogeneous placement is achieved to increase the approximation quality where needed and to emphasize regions of higher importance.

The procedure is very simple to implement, extremely stable, and efficiently provides approximations even for high-dimensional state spaces.

The results provided in this paper can immediately be used for recursive nonlinear filtering of multi-dimensional nonlinear stochastic dynamic systems. For that purpose, the posterior density resulting from a certain processing step, i.e., prediction step or measurement update, is approximated by a Dirac mixture, which is then used as an approximate but convenient prior for the subsequent processing step.

REFERENCES

- [1] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian Estimation using Gaussian Sum Approximations," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, Aug. 1972.
- [2] S. Challa, Y. Bar-Shalom, and V. Krishnamurthy, "Nonlinear Filtering via Generalized Edgeworth Series and Gauss-Hermite Quadrature," *IEEE Transactions on Signal Processing*, vol. 48, no. 6, pp. 1816–1820, June 2000.
- [3] D. Brigo, B. Hanzon, and F. LeGland, "A Differential Geometric Approach to Nonlinear Filtering: The Projection Filter," *IEEE Transactions on Automatic Control*, vol. 42, no. 2, pp. 247–252, 1998.
- [4] N. Oudjane and S. Rubenthaler, "Stability and Uniform Particle Approximation of Nonlinear Filters in Case of Non-Ergodic Signal," Prépublication, Laboratoire de Probabilités et Modèles Aléatoires, Université de Paris VI, Tech. Rep. PMA786, Jan. 2003. [Online]. Available: <http://www.proba.jussieu.fr/mathdoc/textes/PMA-786.pdf>
- [5] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [6] E. Bølviken, P. J. Acklam, N. Christophersen, and J.-M. Størdal, "Monte Carlo Filters for Non-Linear State Estimation," *Automatica*, vol. 37, pp. 177–183, 2001.
- [7] A. Doucet, S. Godsill, and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [8] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 2, pp. 79–86, 1951.
- [9] D. H. Johnson and S. Sinanovic, "Symmetrizing the Kullback–Leibler Distance," submitted to *IEEE Signal Processing Letters*, 2002.
- [10] O. C. Schrempf, D. Brunn, and U. D. Hanebeck, "Density Approximation Based on Dirac Mixtures with Regard to Nonlinear Estimation and Filtering," in *Proceedings of the 2006 IEEE Conference on Decision and Control (CDC 2006)*, San Diego, California, Dec. 2006, pp. 1709–1714.
- [11] —, "Dirac Mixture Density Approximation Based on Minimization of the Weighted Cramér-von Mises Distance," in *Proceedings of the 2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*, Heidelberg, Germany, Sept. 2006, pp. 512–517.
- [12] O. C. Schrempf and U. D. Hanebeck, "Recursive Prediction of Stochastic Nonlinear Systems Based on Dirac Mixture Approximations," in *Proceedings of the 2007 American Control Conference (ACC 2007)*, New York, New York, July 2007, pp. 1768–1774.
- [13] U. D. Hanebeck and O. Feiermann, "Progressive Bayesian Estimation for Nonlinear Discrete-Time Systems: The Filter Step for Scalar Measurements and Multidimensional States," in *Proceedings of the 2003 IEEE Conference on Decision and Control (CDC 2003)*, Maui, Hawaii, Dec. 2003, pp. 5366–5371.
- [14] O. C. Schrempf and U. D. Hanebeck, "A State Estimator for Nonlinear Stochastic Systems Based on Dirac Mixture Approximations," in *Proceedings of the 4th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2007)*, vol. SPSMC, Angers, France, May 2007, pp. 54–61.
- [15] E. C. Anderson, "Monte Carlo Methods and Importance Sampling," *Lecture Notes for Statistical Genetics*, vol. 578C, 1999. [Online]. Available: http://ib.berkeley.edu/labs/slatkin/eriq/classes/guest_lect/mc_lecture_notes.pdf
- [16] W.-L. Loh, "On Latin Hypercube Sampling," *The Annals of Statistics*, vol. 24, no. 5, pp. 2058–2080, 1996.
- [17] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, 1992.
- [18] —, "Point Sets and Sequences with Small Discrepancy," *Monatshefte für Mathematik*, vol. 104, pp. 273–337, 1987.
- [19] J. Spanier and E. H. Maize, "Quasi-Random Methods for Estimating Integrals Using Relatively Small Samples," *SIAM Review*, vol. 36, no. 1, pp. 18–44, March 1994.
- [20] D. E. Knuth, *The Art of Computer Programming*, 3rd ed. Addison-Wesley, 1998, vol. 2 / Seminumerical Algorithms.