University of West Bohemia
Faculty of Applied Sciences
Department of Cybernetics

MASTER'S THESIS

PILSEN, 2020 Jakub Matoušek

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2019/2020

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Jakub MATOUŠEK**
Osobní číslo:	**A18N0057P**
Studijní program:	**N3918 Aplikované vědy a informatika**
Studijní obor:	**Kybernetika a řídicí technika**
Téma práce:	**Metoda bodových mas v úloze odhadu stavu a navigace**
Zadávající katedra:	**Katedra kybernetiky**

Zásady pro vypracování

1) Seznamte se s metodou bodových mas v úloze odhadu stavu.
2) Analyzujte vliv volby mřížky na kvalitu odhadu a výpočetní náročnost metody bodových mas.
3) Navrhněte řešení pro návrh mřížky, která povedou na kvalitnější odhad stavu při zachování výpočetní náročnosti.
4) Implementujte metodu bodových mas a navržená řešení pro návrh mřížky v prostředí MATLAB a v rámci toolboxu „Nonlinear Estimation Framework".
5) Ověřte navržené algoritmy v úloze terénní navigace.
6) Zhodnoťte dosažené výsledky.

Rozsah diplomové práce: **40-50**
Rozsah grafických prací: **dle potřeby**
Forma zpracování diplomové práce: **tištěná**

Seznam doporučené literatury:

[1] N. Bergman, „Recursive Bayesian estimation: navigation and tracking applications“, Ph.D. thesis, 1999.
[2] J. Královec, M. Šimandl, „Filtering, prediction and smoothing with point-mass approach“, IFAC Proceedings Volumes, Volume 37, Issue 6, 2004, Pages 375-380.
[3] N. Sirola, „Nonlinear filtering with piecewise probability densities“, Ph.D. thesis, 2007.
[4] H. W. Sorenson, „On the development of practical nonlinear filters“, Information Sciences 7, 1974, Pages 230-270.
[5] M. Šimandl, J. Královec, T. Söderström, „Advanced point-mass method for nonlinear state estimation“, Automatica 42(7), 2006, Pages 1133-1145.

Vedoucí diplomové práce: **Ing. Jindřich Duník, Ph.D.**
Katedra kybernetiky

Datum zadání diplomové práce: **1. října 2019**
Termín odevzdání diplomové práce: **25. května 2020**

Západočeská univerzita v Plzni
L.S.
Fakulta aplikovaných věd

Doc. Dr. Ing. Vlasta Radová
děkanka

Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

V Plzni dne 1. října 2019

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplmovou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne May 7, 2020

. . . . . .
vlastnoruční podpis

Declaration

I hereby declare that I have done this thesis independently, all texts in this thesis are original, and all the sources have been quoted and acknowledged.

Abstract

This diploma thesis deals with state estimation using Point-Mass Filter. Mainly two topics are discussed; first, a two novel designs of unequally located grid points resulting in improved estimation accuracy with the same computational complexity, second, efficient implementation of the filter in the publicly available MATLAB toolbox called Nonlinear Estimation Framework. Part of the thesis is also focused on the analysis of error while approximating asymmetrical probability density functions by point-mass density. Lastly, the implemented point-mass filter is validated and compared against other filters by numerical simulations for various systems including terrain aided navigation type scenarios.

Keywords: state estimation, nonlinear filtering, nonlinear system, point-mass method

Anotace

Tato diplomová práce se zabývá problémem odhadu stavu použitím metody bodových mas. Předně jsou zde diskutovány dvě témata; zaprvé dva nové typy mřížky s nerovnoměrně rozdělenými body, jež zlepšuje přesnost odhadu při zachování stejné výpočetní náročnosti a za druhé efektivní implementace ve veřejně dostupném MATLAB toolboxu "Nonlinear Estimation Framework". Část této práce se také věnuje analýze chyby aproximace hustotních funkcí pomocí hustotních funkcí bodových mas. Nakonec je implementovaná metoda bodových mas ověřena a porovnána s jinými filtry sérií simulací pro odhad stavu různých systému včetně ukázky aplikace věnované terénní navigaci.

Klíčová slova: odhadu stavu, nelineární filtrace, nelineární systém, metoda bodových mas

Poděkování

Rád bych poděkoval Ing. Jindřichu Duníkovi, Ph.D. za čas, cenné rady, pomoc a neutuchající pozitivitu při zpracovávání této diplomové práce, ostatních článků a konferenčních příspěvků. Také bych rád poděkoval Ing. Ivo Punčocháři, Ph.D. a Bc. Janu Krejčímu za připomínky k této práci. V neposlední řadě bych rád poděkoval celé skupině IDM v čele s Doc. Ing. Ondřejem Strakou, Ph.D. za příjemnou atmosféru, možnost podílet se na publikační činnosti skupiny a možnost zůčastni se konference.

Accompanying literature

[1] Matoušek, J. and Duník, J. and Straka, O. Point-mass fitler: Density specific grid design and implementation. *15th European Workshop on Advanced Control and Diagnosis.* Bologna, Italy, 2019.

[2] Duník, J. and Straka, O. and Matoušek, J. Conditional density driven grid design in point-mass filter. *45th International Conference on Acoustics, Speech, and Signal Processing.* Barcelona, Spain, 2020.

[3] Matoušek, J. and Duník, J. and Straka, O. Density Difference Grid Design in Point-Mass Filter *Advanced Control Design and Fault Diagnosis, special issue of Energies (ISSN 1996-1073)* MDPI. - **In review**

Contents

Nomenclature

Abbreviations

BRR	Bayesian recursive relations
DDG	Density difference grid
DSG	Density specific grid
GD	Gaussian distribution
IE	Integral error
IMU	Inertial mesurement unit
INS	Inerital navigation system
KF	Kalman filter
MSE	Means square error
NEF	Nonlinear Estimation Framework (see http://nft.kky.zcu.cz/nef)
PDF	Probability density function
PF	Particle filter
PMD	Point-mass density

PMF	Point-mass filter
PVA	Position, Velocity and Acceleration
RV	Random variable
STD	Standard deviation
TAN	Terrain-aided navigation
UKF	Unscented Kalman filter

Algorithms Defining Variables

σ_{in}	Number of STDs covered by the denser grid
σ_{out}	Number of STDs covered by the whole grid
N	number of grid points

Math Symbols and Operations

$\mathsf{cov}(\mathbf{w})$	covariance of random variable $\mathbf{w}$
$\mathsf{det}\,(\mathbf{P})$	determinant of $\mathbf{P}$
$\mathsf{diag}(\mathbf{P})$	makes all non diagonal elements of $\mathbf{P}$ equal to 0
Γ	Gamma function
$\mathbf{x}(j)$	j-th element of the vector $\mathbf{x}$
$\mathsf{E}(\mathbf{w})$	mean of random variable $\mathbf{w}$

Chapter 1

Introduction

State estimation of nonlinear discrete-time stochastic dynamic systems from noisy or incomplete measurements has been a subject of considerable research interest for the last several decades. This topic plays an important role in various fields such as navigation, speech and image processing, fault detection, and adaptive or optimal control. As the development of autonomous vehicles or aircraft of all types is a key interest for plenty of renowned companies like Google, Tesla or Honeywell, the investment into research of problems tied to these self-driving vehicles is immense. One of the key information for autonomous applications is knowledge of the position of the controlled object, therefore nearly all automation is tied to navigation problems.

1.1 Terrain-Aided Navigation

The vehicle position, velocity and attitude (PVA) is usually determined based on its initial PVA and measurements from the inertial measurement unit (IMU). The IMU consists of gyroscopes and accelerometers that are used to measure linear accelerations and angular velocities of vehicles. IMU with a processing unit forms the inertial navigation system (INS). INS provides dead-reckoned estimates of vehicle PVA [9]. The processing unit is configured to process inertial measurement and initial navigation state to estimate navigational information.

Unfortunately, because of errors in inertial measurements, the inertial navigation information diverges in time. The rate of the divergence is driven by the grade (i.e. accuracy) of

the IMU. To reduce the divergence the INS needs to be frequently re-initiated. The initialization can be done using a variety of absolute sensors, like GPS receiver, magnetometer or odometer. But as such signals come from the outside of the aircraft, they can be jammed by a hostile entity, the satellite or sensor can be damaged, etc. The safer solution is to use information that is not radiated from outside, such as prerecorded terrain maps. The terrain-aided navigation (TAN) utilizes a measurement of a terrain altitude which is below the vehicle. The terrain altitude can be measured by various altimeters. This measurement is then correlated with the horizontal positions via the map, to re-initialize the INS. The base for all correlating algorithms is the method of state estimation.[3, 16]

1.2 State Estimation

Following the Bayesian approach, a general solution to the state estimation is given by the Bayesian recursive relations (BRRs) that are used to compute the probability density functions (PDFs) of the state (in navigation application represented by PVA) conditioned on the measurements. The conditional PDFs provide a full description of the immeasurable state of a nonlinear or non-Gaussian stochastic dynamic system (i.e. of a navigated vehicle). The relations are, however, exactly solvable for only a limited set of models for which the linearity is usually a common factor. This class of exact Bayesian estimators is represented e.g., by the Kalman filter (KF). In other cases, an approximate solution to the BRRs has to be employed. These approximate filtering methods can be divided with respect to the validity of the resulting estimates into global and local filters [21, 23].

The *local* filters provide computationally efficient estimates predominantly in the form of the conditional mean and covariance matrix with potentially limited performance due to the inherent underlying Gaussianity assumption, which is not always realistic. It should be also noted that the first two moments usually can not fully describe the immeasurable state. These filters are represented by the extended Kalman filter, the unscented Kalman filter, the cubature Kalman filter, stochastic integration filter, etc. [1, 11, 23, 6, 19].

As opposed to the local filters, the *global* filters provide estimates in the form of conditional PDFs without any assumptions on the conditional distribution family. The global filters are capable of estimating the state of a strongly nonlinear or non-Gaussian system but usually at the cost of higher computational demands. Among these, the Gaussian sum filter [22], the particle filter [4], and the point-mass filter [24] have attracted a considerable attention [8].

In this thesis, the point-mass filter (PMF) is considered and further treated. The PMF is based on a numerical solution to the BRRs using deterministic grid-based numerical integration rules, which means the BRRs are solved and the conditional PDF is computed at the grid points only. Although a conceptual design of the PMF is known from the seventies [21], a TAN application of the PMF is reported as late as nineties [3] mainly because of the computational complexity of the filter. Therefore, since the development of the PMF, extensive attention is devoted to reducing the computational complexity of the filter by efficient design. In the literature, two main ways of improving effectiveness can be found;

- Design of the Rao-Blackwellised (or marginalized) PMF [8, 26],
- Efficient convolution in the PMF prediction step [3, 24, 25, 20, 27, 18].

The former approach is based on the decomposition of the state vector into two parts; nonlinearly and linearly modeled. As a consequence, just a nonlinearly modeled part of the state is estimated by the computationally expensive PMF and the remaining part of the state vector can be estimated by a set of computationally cheap KFs. The Rao-Blackwellised PMF is thus suitable for possibly high-dimensional models with *specific* conditionally linear structure and Gaussian noises.

The latter approach *typically* does not restrict the class of the allowed state-space models, but it rather focuses on a computationally efficient implementation of the convolution, in the prediction of the PMF. The PMFs with computationally efficient convolution, using fixed or time-varying number of grid points, assume *equidistantly* spaced grid points, i.e., each grid point has the same vicinity, where the value of the PDF is assumed to be constant. However, the equidistant layout of the points (i.e., an equidistant approximation of the PDF support) may not result in an optimal balance between the computational complexity and accuracy as with the PDF support that has

- A higher volume of the conditional PDF (typically concentrated around the mean), covered by a denser grid to reasonably well capture the PDF shape,
- A lower volume of the conditional PDF (typically at the tail of the distribution), covered by a sparser grid, as the tail is typically flatter.

1.3 Thesis Goal and Summary

The main goal of the thesis is, thus, to design the PMF with a conceptually new grid respecting conditional PDF support and volume. That means the grid is designed to be dense in the PDF support with a high volume of the conditional PDF and sparse otherwise. The proposed grid design is further denoted as the *density specific grid design* (DSG). Additionally, the proposed PMF is also implemented within the Nonlinear Estimation Framework (NEF), which is a publicly available MATLAB toolbox [10]. The implementation was thoroughly tested by comparing implemented PMF with other filters and comparing PMF that is using standard grid design with PMF using DSG design. The DSG design itself was also tested in two TAN scenarios.

The rest of the thesis is organized as follows. Chapter 2 is devoted to an overview of the Bayesian state estimation and mainly to the PMF algorithm itself, its derivation and PMD moments calculation. Chapter 3 describes and compares the standard and proposed grid design. After that, Chapter 4 analyses errors made by approximating asymmetrical PDF using point-mass density. Chapter 5 proposes enhancements to the DSG design and offers alternatives. Chapter 6 deals with the NEF implementation. Numerical illustrations and results are then given in Chapter 7. Chapter 8 concludes the thesis.

Chapter 2

Point-Mass Filter

2.1 System Description

For the purpose of the theoretical part of this thesis consider the following discrete time state-space model of a stochastic dynamic system with additive noises

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k, k = 0, 1, 2, \ldots, T, \tag{2.1}$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, k = 0, 1, 2, \ldots, T, \tag{2.2}$$

where the vectors $\mathbf{x}_k \in \mathbb{R}^{n_x}$, $\mathbf{u}_k \in \mathbb{R}^{n_u}$, and $\mathbf{z}_k \in \mathbb{R}^{n_z}$ represent the *unknown* state of the system and the *known* input and measurement at time instant k, respectively. The state and measurement functions $\mathbf{f}_k : \mathbb{R}^{n_x \times n_u} \to \mathbb{R}^{n_x}$ and $\mathbf{h}_k : \mathbb{R}^{n_x} \to \mathbb{R}^{n_z}$ are supposed to be *known*.

Particular realizations of the state and measurement noises $\mathbf{w}_k$ and $\mathbf{v}_k$ are *unknown*, but their PDFs, i.e., the state noise PDF $p(\mathbf{w}_k)$ and the measurement noise PDF $p(\mathbf{v}_k)$, are supposed to be *known* and independent of the *known* initial state random variable $\mathbf{x}_0$.

2.2 Bayesian Estimation

The point-mass filter is based on the numerical solution to the state estimation which is derived from the Bayesian recursive relations [1]

$$p(\mathbf{x}_k|\mathbf{z}^k) = \frac{p(\mathbf{x}_k, \mathbf{z}_k|\mathbf{z}^{k-1})}{p(\mathbf{z}_k|\mathbf{z}^{k-1})} = \frac{p(\mathbf{x}_k|\mathbf{z}^{k-1})p(\mathbf{z}_k|\mathbf{x}_k)}{p(\mathbf{z}_k|\mathbf{z}^{k-1})}, \tag{2.3}$$

$$p(\mathbf{x}_k|\mathbf{z}^{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}^{k-1})d\mathbf{x}_{k-1}, \tag{2.4}$$

where $p(\mathbf{x}_k|\mathbf{z}^{k-1})$ is the one-step predictive PDF computed by the Chapman-Kolmogorov equation (2.4) and $p(\mathbf{x}_k|\mathbf{z}^k)$ is the filtering PDF computed by the Bayes' rule (2.3). The PDFs $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{z}_k|\mathbf{x}_k)$ are the state transition PDF obtained from system dynamics (2.1) and the measurement PDF obtained from measurement equation (2.2), respectively. The PDF

$$p(\mathbf{z}_k|\mathbf{z}^{k-1}) = \int p(\mathbf{x}_k|\mathbf{z}^{k-1})p(\mathbf{z}_k|\mathbf{x}_k)d\mathbf{x}_k \tag{2.5}$$

is the one-step predictive PDF of the measurement. The estimate of the state is given by the filtering and the predictive PDFs. The recursion (2.3), (2.4) starts from the initial PDF, i.e., $p(\mathbf{x}_0|\mathbf{z}^{-1}) = p(\mathbf{x}_0)$.

Considering the model (2.1), (2.2), the state transition PDF and the BRRs (2.3)–(2.5) should be conditioned also on *available* sequence of the input $\mathbf{u}_k, \forall k$. However, for the sake of notational simplicity, the input signal is assumed to be implicitly part of the condition and it is not explicitly stated, i.e., $p(\mathbf{x}_{k+1}|\mathbf{x}_k) = p(\mathbf{x}_{k+1}|\mathbf{x}_k; \mathbf{u}_k)$, $p(\mathbf{x}_k|\mathbf{z}^k) = p(\mathbf{x}_k|\mathbf{z}^k; \mathbf{u}^{k-1})$, and $p(\mathbf{x}_{k+1}|\mathbf{z}^k) = p(\mathbf{x}_{k+1}|\mathbf{z}^k; \mathbf{u}^k)$.

2.3 Point-mass Density Approximation

The main idea of the PMF is the approximation of conditional PDF $p(\mathbf{x}_k|\mathbf{z}^m)$, where $m = k$ or $m = k-1$ (given by (2.3) and (2.4)), by a *piece-wise constant* point-mass density (PMD) $\hat{p}(\mathbf{x}_k|\mathbf{z}^m; \boldsymbol{\xi}_k)$ defined at the set of the user-defined discrete grid points $\boldsymbol{\xi}_k = \{\boldsymbol{\xi}_k^{(i)}\}_{i=1}^N, \boldsymbol{\xi}_k^{(i)} \in \mathbb{R}^{n_x}$, as follows [5]

$$p(\mathbf{x}_k|\mathbf{z}^m) \approx \hat{p}(\mathbf{x}_k|\mathbf{z}^m; \boldsymbol{\xi}_k) \triangleq \sum_{i=1}^{N} P_{k|m}(\boldsymbol{\xi}_k^{(i)}) S\{\mathbf{x}_k; \boldsymbol{\xi}_k^{(i)}, \boldsymbol{\Delta}_k\}, \tag{2.6}$$

with

- $P_{k|m}(\boldsymbol{\xi}_k^{(i)}) = c_k \tilde{P}_{k|m}(\boldsymbol{\xi}_k^{(i)})$, where $\tilde{P}_{k|m}(\boldsymbol{\xi}_k^{(i)}) = p(\boldsymbol{\xi}_k^{(i)}|\mathbf{z}^m)$ is the value of the conditional PDF $p(\mathbf{x}_k|\mathbf{z}^m)$ evaluated at i-th grid point $\boldsymbol{\xi}_k^{(i)}$, $c_k = \delta_k \sum_{i=1}^N \tilde{P}_{k|m}(\boldsymbol{\xi}_k^{(i)})$ is a normalisation constant, and δ_k is a volume of the i-th point neighbourhood defined below,

- $\boldsymbol{\Delta}_k = [\boldsymbol{\Delta}_k(1), \boldsymbol{\Delta}_k(2), \ldots, \boldsymbol{\Delta}_k(n_x)]^T$ defines a (hyper-)rectangular neighbourhood of a grid point $\boldsymbol{\xi}_k^{(i)}$, where the PDF $p(\mathbf{x}_k|\mathbf{z}^m)$ is assumed to be constant and has value $P_{k|m}(\boldsymbol{\xi}_k^{(i)})$, and

- $S\{\mathbf{x}_k; \boldsymbol{\xi}_k^{(i)}, \boldsymbol{\Delta}_k\}$ is the *selection* function defined as

$$S\{\mathbf{x}_k; \boldsymbol{\xi}_k^{(i)}, \boldsymbol{\Delta}_k\} = 1, \tag{2.7}$$

if $\mathbf{x}_k(j) \in [\boldsymbol{\xi}_k^{(i)}(j) - \frac{\boldsymbol{\Delta}_k(j)}{2}, \boldsymbol{\xi}_k^{(i)}(j) + \frac{\boldsymbol{\Delta}_k(j)}{2}]$ for $j = 1, 2, \ldots, n_x$, and

$$S\{\mathbf{x}_k; \boldsymbol{\xi}_k^{(i)}, \boldsymbol{\Delta}_k\} = 0, \tag{2.8}$$

otherwise, so that

$$\begin{aligned} \int S\{\mathbf{x}_k; \boldsymbol{\xi}_k^{(i)}\} d\mathbf{x}_k &= \boldsymbol{\Delta}_k(1) \times \boldsymbol{\Delta}_k(2) \times \ldots \times \boldsymbol{\Delta}_k(n_x) \\ &= \delta_k. \end{aligned} \tag{2.9}$$

Selection function's S property is that the neighbourhoods $\boldsymbol{\Delta}$ of any points does not overlap. Note that in (2.6)–(2.9) the notation $\mathbf{x}(j)$ meaning the j-th element of the vector $\mathbf{x}$ was used.

The $\boldsymbol{\Delta}_k$ should be for non-equidistant grids examined in this thesis point dependent, therefore notation should be $\boldsymbol{\Delta}_k^{(i)}$, for the sake of simplicity this notation is omitted in the theoretical part of this thesis. Also the number of points N can generally be time dependent N_k, but this case is not dealt with in this thesis.

Illustration of point-mass PDF approximation (2.6) with omitted time indices is shown in Fig. 2.1.

From the theoretical point of view the point-mass approximation considered in this thesis can be viewed as a sum of rescaled uniform distributions. In the literature, however, another interpretation can be found as well, which is based on a sum Dirac delta functions located at the grid points [25]. Relevant discussion considering this topic can be found in [8].

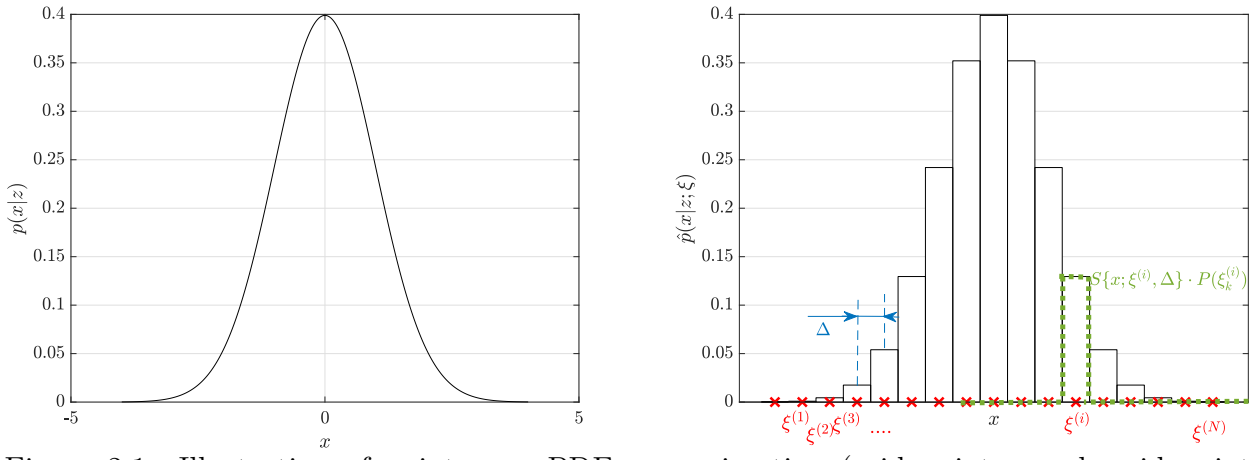

Figure 2.1: Illustration of point-mass PDF approximation (grid points - red, grid point neighbourhood - blue, scaled selection function - green).

2.4 Point-Mass Solution to Bayesian Recursive Relations

Other filtration algorithms are based on assumptions about the estimated system (e.g. linearity, gaussianity, etc.) or approximations. Introducing these restrictions makes it possible to find an analytical (but approximate) solution to the BRRs. On the other hand, the point-mass filter is based on their numerical solution using PMD approximation [5].

2.4.1 Derivation of Measurement-Update

Derivation of a PMF measurement-update starts with BRR for posterior density, [5]

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}^k) &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}^{k-1})}{p(\mathbf{z}_k|\mathbf{z}^{k-1})} \\ &= \tilde{c}_k^{-1} p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}^{k-1}), \end{aligned} \tag{2.10}$$

where $p(\mathbf{x}_k|\mathbf{z}^{k-1})$ is the predictive density from the last step or from initialization, $\tilde{c}_k$ is a normalization constant and

$$p(\mathbf{z}_k|\mathbf{x}_k) = p_{\mathbf{v}_k}(\mathbf{z}_k - \mathbf{h}_k(\mathbf{x}_k)) \tag{2.11}$$

is the PDF of a measurement, which is given by (2.2). For example with Gaussian measurement noise

$$p(\mathbf{z}_k|\mathbf{x}_k) = N(\mathbf{z}_k; \mathbf{h}_k(\mathbf{x}_k), \mathsf{cov}(\mathbf{w}_k)). \tag{2.12}$$

However, only the point-mass approximation of the predictive density is known

$$\hat{p}(\mathbf{x}_k|\mathbf{z}^{k-1}) = \sum_{i=1}^{N} P_{k|k-1}(\boldsymbol{\xi}_k^{(i)}) S\{\mathbf{x}_k : \boldsymbol{\xi}_k^{(i)}, \boldsymbol{\Delta}_k\}. \tag{2.13}$$

Using (2.10)

$$\hat{p}(\mathbf{x}_k|\mathbf{z}^k) = \hat{\tilde{c}}_k^{-1} p(\mathbf{z}_k|\mathbf{x}_k) \sum_{i=1}^{N} P_{k|k-1}(\boldsymbol{\xi}_k^{(i)}) S\{\mathbf{x}_k : \boldsymbol{\xi}_k^{(i)}, \boldsymbol{\Delta}_k\}, \tag{2.14}$$

where $\hat{\tilde{c}}_k$ is an approximation of the normalization constant.

Because of the predictive piece-wise constant approximation of the PDF, the measurement PDF can be also evaluated at $\boldsymbol{\xi}_k^{(i)}$ points only [1]. Therefore, point-mass approximation of the measurement-update probability is

$$\hat{p}(\mathbf{x}_k|\mathbf{z}^k) = \sum_{i=1}^{N} P_{k|k}(\boldsymbol{\xi}_k^{(i)}) S\{\mathbf{x}_k : \boldsymbol{\xi}_k^{(i)}, \boldsymbol{\Delta}_k\}, \tag{2.15}$$

where the value of the filtration PDF at $\boldsymbol{\xi}_k^{(i)}$ is

$$P_{k|k}(\boldsymbol{\xi}_k^{(i)}) = \hat{\hat{c}}_k^{-1} p(\mathbf{z}_k|\mathbf{x}_k = \xi_k^{(i)}) P_{k|k-1}(\boldsymbol{\xi}_k^{(i)}). \tag{2.16}$$

The normalization constant equals

$$\hat{\hat{c}}_k = \int \sum_{i=1}^{N} p(\mathbf{z}_k|\mathbf{x}_k = \xi_k^{(i)}) P_{k|k-1}(\boldsymbol{\xi}_k^{(i)}) S\{\mathbf{x}_k : \boldsymbol{\xi}_k^{(i)}, \boldsymbol{\Delta}_k\} d\mathbf{x}_k. \tag{2.17}$$

Note that, in this case, the grid support stays the same during measurement-update.

2.4.2 Derivation of Time-Update

Predictive probability is given by the Chapman-Kolmogorov equation (2.4) [5]

$$p(\mathbf{x}_{k+1}|\mathbf{z}^k) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}^k) d\mathbf{x}_k, \tag{2.18}$$

where $p(\mathbf{x}_k|\mathbf{z}^k)$ is the filtration density and

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k) = p_{\mathbf{w}_k}(\mathbf{x}_{k+1} - \mathbf{f}_k(\mathbf{x}_k)) \tag{2.19}$$

is a transition PDF of the system, which is given by the dynamics (2.1). Only the approximate filtration density is known (see previous subsection), therefore

$$\hat{p}(\mathbf{x}_{k+1}|\mathbf{z}^k) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k) \sum_{i=1}^{N} P_{k|k}(\boldsymbol{\xi}_k^{(i)}) S\{\mathbf{x}_k : \boldsymbol{\xi}_k^{(i)}, \boldsymbol{\Delta}_k\} d\mathbf{x}_k. \tag{2.20}$$

In order to solve this relation in a PMF philosophy a new grid $\{\boldsymbol{\xi}_{k+1}^{(j)}\}_{j=1}^{N}$ has to be defined. This grid should cover the part of the state-space, where the predictive probability is expected to be. Now the $p(\mathbf{x_{k+1}}|\mathbf{x}_k)$ can be approximated as

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k) \approx \hat{p}(\mathbf{x}_{k+1}|x_k) = \sum_{j=1}^{N} p(\boldsymbol{\xi}_{k+1}^{(j)}|\mathbf{x}_k) S\{\mathbf{x}_{k+1} : \boldsymbol{\xi}_{k+1}^{(j)}, \boldsymbol{\Delta}_{k+1}\}. \tag{2.21}$$

[1] It could be evaluated at different grid points, but then filtering step is computationally more demaming

The time-update probability can be then expressed as

$$\begin{aligned}
\hat{p}(\mathbf{x}_{k+1}|\mathbf{z}^k) &\approx \int \hat{p}(\mathbf{x}_{k+1}|\mathbf{x}_k) \sum_{i=1}^{N} P_{k|k}(\boldsymbol{\xi}_k^{(i)}) S\{\mathbf{x}_k : \boldsymbol{\xi}_k^{(i)}, \boldsymbol{\Delta}_k\} d\mathbf{x}_k \\
&= \int \sum_{j=1}^{N} p(\boldsymbol{\xi}_{k+1}^{(j)}|\mathbf{x}_k) S\{\mathbf{x}_{k+1} : \boldsymbol{\xi}_{k+1}^{(j)}, \boldsymbol{\Delta}_{k+1}\} \sum_{i=1}^{N} P_{k|k}(\xi_k^{(i)}) S\{\mathbf{x}_k : \boldsymbol{\xi}_k^{(i)}, \Delta_k\} d\mathbf{x}_k \\
&\approx \int \sum_{j=1}^{N} \sum_{i=1}^{N} P_{k|k}(\boldsymbol{\xi}_k^{(i)}) p(\boldsymbol{\xi}_{k+1}^{(j)}|\mathbf{x}_k = \boldsymbol{\xi}_k^{(i)}) S\{\mathbf{x}_k : \boldsymbol{\xi}_k^{(i)}, \boldsymbol{\Delta}_k\} S\{\mathbf{x}_{k+1} : \boldsymbol{\xi}_{k+1}^{(j)}, \boldsymbol{\Delta}_{k+1}\} d\mathbf{x}_k \\
&= \sum_{i=1}^{N} \sum_{j=1}^{N} P_{k|k}(\boldsymbol{\xi}_k^{(i)}) p(\boldsymbol{\xi}_{k+1}^{(j)}|\mathbf{x}_k = \boldsymbol{\xi}_k^{(i)}) S\{\mathbf{x}_{k+1} : \boldsymbol{\xi}_{k+1}^{(j)}, \boldsymbol{\Delta}_{k+1}\} \underbrace{\int S\{\mathbf{x}_k : \boldsymbol{\xi}_k^{(i)}, \boldsymbol{\Delta}_k\} d\mathbf{x}_k}_{\delta_k} \\
&= \sum_{j=1}^{N} \underbrace{\sum_{i=1}^{N} P_{k|k}(\boldsymbol{\xi}_k^{(i)}) p(\boldsymbol{\xi}_{k+1}^{(j)}|\mathbf{x}_k = \boldsymbol{\xi}_k^{(i)}) \delta_k}_{P_{k+1|k}(\boldsymbol{\xi}_{k+1}^{(j)})} S\{\mathbf{x}_{k+1} : \boldsymbol{\xi}_{k+1}^{(j)}, \boldsymbol{\Delta}_{k+1}\} \\
&= \sum_{j=1}^{N} P_{k+1|k}(\boldsymbol{\xi}_{k+1}^{(j)}) S\{\mathbf{x}_{k+1} : \boldsymbol{\xi}_{k+1}^{(j)}, \boldsymbol{\Delta}_{k+1}\}, \qquad (2.22)
\end{aligned}$$

where the predictive probability value at a new grid point $\boldsymbol{\xi}_{k+1}^{(j)}$ is

$$P_{k+1|k}(\boldsymbol{\xi}_{k+1}^{(j)}) = \sum_{i=1}^{N} p(\boldsymbol{\xi}_{k+1}^{(j)}|\mathbf{x}_k = \boldsymbol{\xi}_k^{(i)}) P_{k|k}(\boldsymbol{\xi}_k^{(i)}) \delta_k, \forall j. \qquad (2.23)$$

The expression (2.22) is the most computationally demanding step of the PMF. Note that the value of $P_{k+1|k}(\boldsymbol{\xi}_{k+1}^{(j)})$ is independent over j, therefore parallel computation can be used to speed up the algorithm.

2.4.3 Point-Mass Filter Algorithm

The basic algorithm of the PMF can be summarized by the following steps [2, 25]:

Algorithm 1: Point-Mass Filter

---

1. **Initialisation**: Determine number of grid points N. Set $k = 0$, construct the initial grid consisting of points $\{\boldsymbol{\xi}_0^{(i)}\}_{i=0}^{N}$, and define the initial point-mass PDF

$$\hat{p}(\mathbf{x}_0|\mathbf{z}^{-1}; \boldsymbol{\xi}_0) = \sum_{i=1}^{N} P_{0|-1}(\boldsymbol{\xi}_0^{(i)}) S\{\mathbf{x}_0; \boldsymbol{\xi}_0^{(i)}, \boldsymbol{\Delta}_0\}, \qquad (2.24)$$

approximating the initial PDF, i.e., $p(\mathbf{x}_0|\mathbf{z}^{-1}) = p(\mathbf{x}_0)$.

2. **Measurement update**: Compute the filtering point-mass PDF

$$\hat{p}(\mathbf{x}_k|\mathbf{z}^k;\boldsymbol{\xi}_k) = \sum_{i=1}^{N} P_{k|k}(\boldsymbol{\xi}_k^{(i)}) S\{\mathbf{x}_k;\boldsymbol{\xi}_k^{(i)},\boldsymbol{\Delta}_k\}, \tag{2.25}$$

where the value of the filtering PDF at the i-th grid point is

$$P_{k|k}(\boldsymbol{\xi}_k^{(i)}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k=\boldsymbol{\xi}_k^{(i)}) P_{k|k-1}(\boldsymbol{\xi}_k^{(i)})}{\sum_{j=1}^{N} p(\mathbf{z}_k|\mathbf{x}_k=\boldsymbol{\xi}_k^{(j)}) P_{k|k-1}(\boldsymbol{\xi}_k^{(j)})\delta_k}. \tag{2.26}$$

3. **Grid construction:** Based on the filtering estimate (2.25) and state equation (2.1), construct the new grid of points $\{\boldsymbol{\xi}_{k+1}^{(j)}\}_{j=1}^{N}$.

4. **Time update**: Compute the predictive point-mass PDF at the new grid of points according to

$$\hat{p}(\mathbf{x}_{k+1}|\mathbf{z}^k;\boldsymbol{\xi}_{k+1}) = \sum_{j=1}^{N} P_{k+1|k}(\boldsymbol{\xi}_{k+1}^{(j)}) S\{\mathbf{x}_{k+1};\boldsymbol{\xi}_{k+1}^{(j)},\boldsymbol{\Delta}_{k+1}\}, \tag{2.27}$$

where the value of the predictive PDF at the j-th grid point is given by the convolution

$$P_{k+1|k}(\boldsymbol{\xi}_{k+1}^{(j)}) = \sum_{i=1}^{N} p(\boldsymbol{\xi}_{k+1}^{(j)}|\mathbf{x}_k = \boldsymbol{\xi}_k^{(i)}) P_{k|k}(\boldsymbol{\xi}_k^{(i)})\delta_k. \tag{2.28}$$

5. Set $k = k + 1$ and go to the step 2).

The time update step is most time demanding operation of the PMF.

2.4.4 Multi-Step Prediction and Smoothing

Multi-Step Prediction

Prediction of the l steps by the point-mass approximation is an estimation of PDF $p(\mathbf{x}_{k+l}|\mathbf{z}^k)$, where $l \in \mathbb{N} \setminus 1$. Multi-step prediction consists of applying time update step (2.27) l times for l step prediction. It means the definition of a new grid and computation of the convolution is performed l times [13].

Smoothing

Smoothing is a state-estimation task of finding more accurate estimates than filtering ones utilizing the knowledge of future measurements. Therefore the smoothing PDF is $p(\mathbf{x}_l|\mathbf{z}^k)$, where k is the step that the last measurement came and $l = 0, 1, ..., k-1$.

Assuming that the filtering $p(\mathbf{x}_l|\mathbf{z}^l)$ and the predictive $p(\mathbf{x}_{l+1}|\mathbf{z}^l)$ PDFs, are known, the general formula for the basic one-step smoothing algorithm is

$$p(\mathbf{x}_l|\mathbf{z}^k) = p(\mathbf{x}_l|\mathbf{z}^l) \int \frac{p(\mathbf{x}_{l+1}|\mathbf{z}^k)}{p(\mathbf{x}_{l+1}|\mathbf{z}^l)} p(\mathbf{x}_{l+1}|\mathbf{x}_l) d\mathbf{x}_{l+1}, \tag{2.29}$$

where $l = k-1$ [14].

Using the point-mass approximation, the smoothing PDF can be computed as

$$\hat{p}(\mathbf{x}_l|\mathbf{z}^k; \boldsymbol{\xi}_k) = \sum_{i=j}^{N_l} P_{l|k}(\boldsymbol{\xi}_l^{(j)}) S\{\mathbf{x}_l; \boldsymbol{\xi}_l^{(j)}, \boldsymbol{\Delta}_l\}, \tag{2.30}$$

where the value of the smoothing PDF at the j-th grid point is

$$P_{l|k}(\boldsymbol{\xi}_l^{(j)}) = P_{l|l}(\boldsymbol{\xi}_l^{(j)}) \sum_{i=1}^{N_l} \frac{P_{l+1|k}(\boldsymbol{\xi}_{l+1}^{(i)})}{P_{l+1|l}(\boldsymbol{\xi}_{l+1}^{(i)})} p(\boldsymbol{\xi}_{l+1}^{(j)}|\mathbf{x}_l = \boldsymbol{\xi}_l^{(i)}), \tag{2.31}$$

where N_l is the number of points in the grid that is supporting PDFs conditioned by z^l. Note that the smoothing PDF uses the same grid as the filtering PDF. Moreover, all conditional PDFs (filtering, predictive, and smoothing) for the time instant k have the same grid support. Furthermore, the $p(\boldsymbol{\xi}_{k+1}^{(j)}|\mathbf{x}_k = \boldsymbol{\xi}_k^{(i)})$ is already known from time update step.

2.4.5 PMD Moments Calculation

Because the PMF filter provides estimates in the form of the point-mass density $\hat{p}(\mathbf{x}_k|\mathbf{z}^m; \boldsymbol{\xi}_k)$, approximating a conditional PDF $p(\mathbf{x}_k|\mathbf{z}^m)$, the relations to compute the moments of the said density are needed for estimates evaluation (e.g. comparison of mean and variance of filtering PDF with other filters), for analysis and also for an advanced grid setup. The moments of the conditional PDF, can be easily approximated as

$$\mathsf{E}[\mathbf{x}_k|\mathbf{z}^m] \approx \hat{\mathbf{x}}_{k|m}(\boldsymbol{\xi}_k) = \sum_{i=1}^{N} \delta_k P_{k|m}(\boldsymbol{\xi}_k^{(i)}) \boldsymbol{\xi}_k^{(i)}, \tag{2.32}$$

$$\mathsf{cov}[\mathbf{x}_k|\mathbf{z}^m] \approx \hat{\mathbf{P}}_{k|m}(\boldsymbol{\xi}_k) = \sum_{i=1}^{N} \delta_k P_{k|m}(\boldsymbol{\xi}_k^{(i)}) \left(\boldsymbol{\xi}_k^{(i)} - \hat{\mathbf{x}}_{k|m}\right) \left(\boldsymbol{\xi}_k^{(i)} - \hat{\mathbf{x}}_{k|m}\right)^T \tag{2.33}$$

and general n-th central moment

$$\boldsymbol{\mu}_{k|m}^{n} \approx \sum_{i=1}^{N} \delta_k P_{k|m}(\boldsymbol{\xi}_k^{(i)}) \left(\boldsymbol{\xi}_k^{(i)} - \hat{\mathbf{x}}_{k|m}\right) \left(\left(\boldsymbol{\xi}_k^{(i)} - \hat{\mathbf{x}}_{k|m}\right)^T\right)^{\otimes(n-1)}, \tag{2.34}$$

where $\otimes$ is a Kronecker product. Derivation of the moments computation can be found e.g. in [8].

Chapter 3

Point-Mass Filter Grid Design

A critical part of the PMF design is a specification of the grid points in time update step (2.24) and, by extension, during initialisation (2.27).

When designing a grid, firstly the number of grid points should be decided. The more grid points N is considered, the better the accuracy of the density approximation can be achieved but at the cost of higher computational demands. Due to the convolution (2.27), (2.28), the computational complexity grows quadratically with N. Thus, the number of grid points is solely driven by the available computational capacity and required precision of the estimate.

The grid should be designed to cover a *significant* region of the conditional PDF support *sufficiently well*. Therefore, it is necessary to specify two grid characteristics

1. Region of the PDF support to be covered by the grid points,
2. Locations of grid points inside the region.

3.1 Standard Grid Design

3.1.1 Region of the PDF Support

Regarding the *first* characteristic, it is relatively easy to specify the covered region $\mathcal{R}$ of the PDF support for the *initial* PDF $p(\mathbf{x}_0|\mathbf{z}^{-1})$. The region is typically rectangular and centered

around the mean $\hat{\mathbf{x}}_{0|-1} = \mathsf{E}[\mathbf{x}_0|\mathbf{z}^{-1}]$ and is selected so that

$$\int_{\mathcal{R}} p(\mathbf{x}_0|\mathbf{z}^{-1}) > P_{\text{thr}}, \tag{3.1}$$

where P_{thr} is a user defined threshold. For the predictive PDF $p(\mathbf{x}_k|\mathbf{z}^{k-1})$, the specification of the region is more involving since the predictive PDF is *not* available at "Grid construction" step of Algorithm 1 (the PDF is to be computed in the subsequent step "Time update"). A reasonable approach is to compute the first two predictive moments of the state

$$\hat{\mathbf{x}}_{\text{A},k+1|k} = \mathsf{E}[\mathbf{x}_{k+1}|\mathbf{z}^k], \tag{3.2}$$

$$\mathbf{P}_{\text{A},k+1|k} = \mathsf{cov}[\mathbf{x}_{k+1}|\mathbf{z}^k], \tag{3.3}$$

and to imagine an approximate Gaussian[1] predictive PDF

$$p_{\text{A}}(\mathbf{x}_{k+1}|\mathbf{z}^k) = \mathcal{N}\{\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{\text{A},k+1|k}, \mathbf{P}_{\text{A},k+1|k}\}, \tag{3.4}$$

where the notation $\mathcal{N}\{\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}\}$ stands for the Gaussian PDF of the random variable[2] $\mathbf{x}$ with the mean $\hat{\mathbf{x}}$ and the covariance matrix $\mathbf{P}$. Then, the sought region $\mathcal{R}$ can be found analogously to (3.1), where the approximate predictive PDF $p_{\text{A}}(\mathbf{x}_{k+1}|\mathbf{z}^k)$ (3.4) is used instead of the initial one $p(\mathbf{x}_0|\mathbf{z}^{-1})$.

The predictive moments (3.2) and (3.3) are computed on the basis of known filtering moments (given by (2.32), (2.33), and (2.25)) and state equation (2.1). If the dynamics is nonlinear, *approximate* predictive moments can be computed e.g., by unscented transformation or various numerical integration rules [11, 7].

σ Ellipse

For a scalar random variable $x \in \mathbb{R}$ with $\mathcal{N}\{x; m, \sigma^2\}$ a region centered around mean value (be called confidence region) can be found so that there is an α probability that the realization will be inside this region as shown on top in Figure 3.1. For 1D case

$$\alpha = P(m - \kappa\sigma \leq x \leq m + \kappa\sigma) = P\left(\frac{(x-m)^2}{\sigma^2} \leq \kappa^2\right) = F_{\chi^2,1}(\kappa^2), \tag{3.5}$$

[1] The approximate PDF (3.4) is used for the region $\mathcal{R}$ determination only. The predictive PDF (2.27) is computed in this region.

[2] Note that, for simplicity, the random variable has same notation as its realization.

where $P(X)$ is a probability measure of set X. For a multidimensional Gaussian distribution the contour line

$$constant = \frac{1}{\sqrt{(2\pi)^{n_x}\det(\mathbf{P})}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{m})^T\mathbf{P}^{-1}(\mathbf{x}-\boldsymbol{m})\right), \tag{3.6}$$

$$constant' = (\mathbf{x}-\boldsymbol{m})^T\mathbf{P}^{-1}(\mathbf{x}-\boldsymbol{m}), \tag{3.7}$$

is an ellipsoid. Then a probability of an event for $n_x \in \mathbb{N}$ dimension normal distribution is

$$\alpha = P\left((\mathbf{x}-\boldsymbol{m})^T\mathbf{P}^{-1}(\mathbf{x}-\boldsymbol{m}) \leq \kappa^2\right) = F_{\chi^2,n_x}(\kappa^2), \tag{3.8}$$

where $\mathbf{P}$ is the covariance matrix. Now $\kappa\sigma$ ellipsoid can be defined by all $\mathbf{x}$ that satisfy (3.8) for constant κ. For 2 dimensions the $\kappa\sigma$ ellipse can be seen at the bottom of Figure 3.1.

Using the mentioned above, the region $\mathcal{R}$ from (3.1) can be, for non-Gaussian distributions, approximately found as an ellipsoid. Note that it is computationally efficient to define the region as a (hyper-)rectangle around the ellipsoid as shown in Figure 6.1. Most standard grid designs take advantage of this and DSG design inherits this shape of the grid too.

Alternative approach would be to use the multidimensional Chebyshev's inequality which states

$$P\left(\sqrt{(\mathbf{x}-\boldsymbol{m})^T\mathbf{P}^{-1}(\mathbf{x}-\boldsymbol{m})} > t\right) \leq \frac{n_x}{t^2}. \tag{3.9}$$

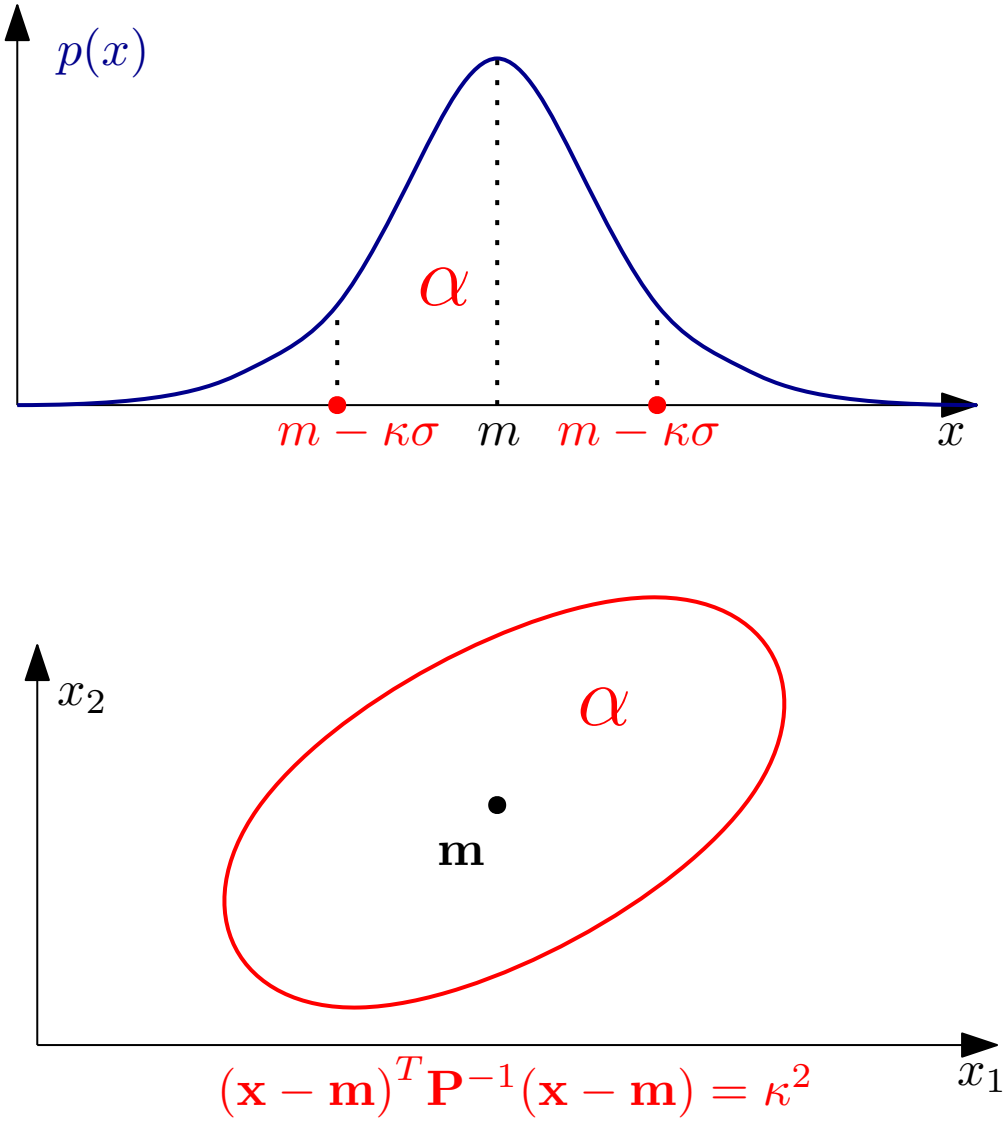

Figure 3.1: PDF, approximation by point-mass density, and approximation error.

3.1.2 Location of Grid Points

Regarding the *second* characteristic, the grid is usually (hyper-)rectangular with *equally* spaced points [3, 8, 18]. However, the equally spaced grid points $\{\boldsymbol{\xi}_k^{(i)}\}_{i=1}^N$ result in a point-mass density with significantly *spatially* varying approximation error defined as

$$\tilde{p}(\mathbf{x}_k|\mathbf{z}^m; \boldsymbol{\xi}_k) = |p(\mathbf{x}_k|\mathbf{z}^m) - \hat{p}(\mathbf{x}_k|\mathbf{z}^m; \boldsymbol{\xi}_k)|, \tag{3.10}$$

where the notation $|\cdot|$ stands for the absolute value.

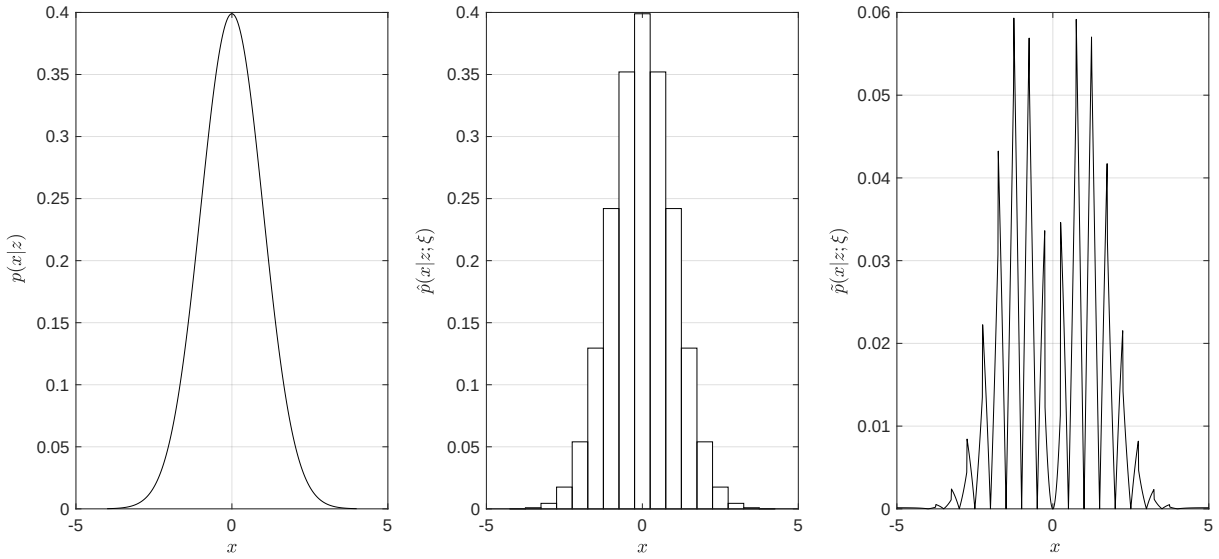

Figure 3.2: PDF, approximation by point-mass density, and approximation error.

The true PDF $p(\mathbf{x}_k|\mathbf{z}^m)$ with its point-mass approximation $\hat{p}(\mathbf{x}_k|\mathbf{z}^m;\boldsymbol{\xi}_k)$, and the approximation error $\tilde{p}(\mathbf{x}_k|\mathbf{z}^m;\boldsymbol{\xi}_k)$ (3.10) are illustrated in Fig. 3.2. The figure clearly shows spatially varying approximation error of the point-mass density; the larger the slope of the PDF is, the larger approximation error is observed. Thus, a large error can be expected in the vicinity of the mean value, whereas rather a low error at the tail of a majority of distributions (i.e., close to the boundary of the grid) as the tail is relatively flat (i.e., almost constant).

Therefore, having a fixed number of grid points N, it might be better to distribute the grids over the approximated PDF support *unequally*; more points in the middle of the grid, less point at the grid boundaries.

3.1.3 Grid Design for Scalar Variable: Illustration

The *standard* approach to the equidistant grid design lies in the following steps:

Algorithm 2: Illustration of Standard Grid Design

(i) Specification of the region $\mathcal{R}$ from (3.1), which is, in the considered scalar case, a line segment, for example, if $P_{\text{thr}} \approx 0.999999$, then the region is a closed line segment of length $\ell = 12$ with endpoints $-6\sigma_{k+1|k} = -6$ and $6\sigma_{k+1|k} = 6$ [17],

(ii) Specification of an allowed number of the grid points N, e.g. $N = 21$,

(iii) Calculation of equidistantly spaced grid points $\{\xi_{k+1}^{(i)}\}_{i=1}^{N}$, i.e., with the grid point vicinity $\Delta_{k+1} = \delta_{k+1} = \frac{\ell}{N-1} = 0.6$ constant for all grid points.

3.2 Density Specific Grid Design

The proposed *density specific grid (DSG) design*, is based on an unequally spaced placement of the fixed number of points over the PDF support to be approximated. The main idea is to approximate the PDF where the volume of the PDF is significant or significantly changing by a denser grid (as in this area, a slope of the PDF is expected to be significant) and the tail of the PDF by a sparser grid (as the tail is flat). The DSG design is illustrated for a scalar variable first and then generalized and discussed for a vector variable.

Even though this idea assumes the approximated PDF is unimodal and symmetric, the DSG design should work for asymmetrical and multimodal distributions as long as the "important"/fast-changing part of the PDF is around its mean value, as illustrated in Figure 3.3. This condition is satisfied in many scenarios. Moreover, an alternative approach to the denser region determination for multimodal PDFs is offered in Section 5.2.

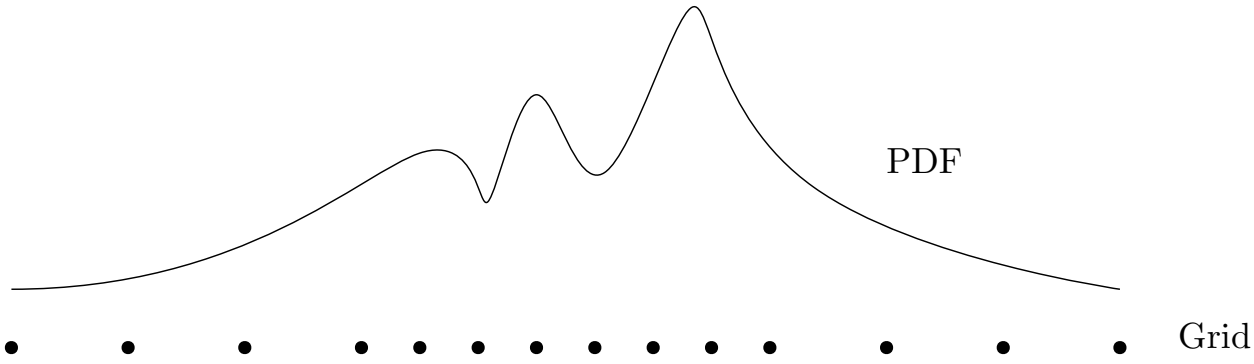

Figure 3.3: Illustration of DSG design approximation of non-gaussian PDF.

3.2.1 Region of the PDF Support

The grid construction step precedes the PMF prediction step. The determination of the PDF support region to be covered by the grid points is only approximate as it is based on the assumption of the symmetric Gaussian predictive PDF (see discussion in Section 3.1.1).

However, it must be noted that the assumption is used just for the *region determination* and *grid point placement*; the predictive PDF itself is *not* restricted by any density-type assumption.

The region $\mathcal{R}$ for the whole grid is sought analogically to the standard grid but in addition, another region $\mathcal{R}_{center}$ for the denser grid is needed. This $\mathcal{R}_{center}$ has the same midpoint but is smaller than $\mathcal{R}$.

3.2.2 Location of Grid Points

Going back to the approximate predictive Gaussian PDF $p_{\mathrm{A}}(x_{k+1}|z^k)$ (3.4) used for predictive PDF region determination and grid construction, let, without loss of generality, an approximate predictive PDF in the form of the standard normal distribution

$$p_{\mathrm{A}}(x_{k+1}|z^k) = \mathcal{N}\{x_{k+1}; 0, 1\}, \tag{3.11}$$

be considered, i.e., the standard deviation is $\sigma_{k+1|k} = \sqrt{P_{k+1|k}} = 1$. The newly proposed *DSG design* leads to non-equidistantly placed grid points, where a denser grid is constructed around the estimated predictive mean value (obtained from e.g., UKF prediction step) vicinity and the sparser grid is used at the tail of the PDF. The DSG design is proposed with *minimal* computational overhead with respect to a standard grid design.

3.2.3 Standard Grid Design for Scalar Variable: Illustration

The creation of the DSG can be summarized by the following algorithm.

Algorithm 3: Illustration of Density Specific Grid Design

(i) Specification of the region $\mathcal{R}$ from (3.1), which is, in the considered scalar case, a line segment. For example, if $P_{\mathrm{thr}} = 0.999999$, then the region is a closed line segment of length $\ell = 16$ with endpoints $-6\sigma_{k+1|k} = -6$ and $6\sigma_{k+1|k} = 6$.

(ii) Splitting region $\mathcal{R}$, defined by the line segment, into two parts; the *central* part $\mathcal{R}_{\mathrm{centre}}$ around the mean value $\hat{x}_{k+1|k} = 0$, where a PDF slope is expected to be significant, and the remaining *tail* part $\mathcal{R}_{\mathrm{tail}}$, where a PDF is expected to be almost flat. For example, the central region $\mathcal{R}_{\mathrm{centre}}$ is defined by a closed line segment of length $\ell = 6$

with endpoints $-3\sigma_{k+1|k} = -3$ and $3\sigma_{k+1|k} = 3$ and the tail region $\mathcal{R}_{\text{tail}}$ is defined by two closed line segments of length $\ell = 3$ with endpoints -6, -3 and 3, 6, respectively.

(iii) Specification of an allowed number of the grid points N. For example, number of the grid points is selected as $N = 21$.

(iv) Allocation of grid points into both subregions $\mathcal{R}_{\text{centre}}$ and $\mathcal{R}_{\text{tail}}$; the former should contain more grid points than the latter one. For example, the subregion $\mathcal{R}_{\text{centre}}$ is covered by $N_{\text{centre}} = \frac{3}{4}N \approx 17$, whereas tail region $\mathcal{R}_{\text{tail}}$ by remaining points $N_{\text{tail}} = \frac{1}{4}N = 4$. As the tail region is given by two parts (at each side of the central region $\mathcal{R}_{\text{centre}}$), each part is covered by 2 points.

(v) Calculation of *non-equidistantly* spaced grid points $\{\xi_{k+1}\}_{i=1}^{N}$ so that the central region $\mathcal{R}_{\text{centre}}$ is equidistantly covered by N_{centre} points and the tail region $\mathcal{R}_{\text{tail}}$ by N_{tail} points. For example, in $\mathcal{R}_{\text{centre}}$ the grid points are close to each other with the vicinity $\Delta_{\text{centre},k+1} = 0.375$, but in $\mathcal{R}_{\text{tail}}$ the grid point vicinity is $\Delta_{\text{tail},k+1} = 1.5$.

Ideally, the whole space would be covered by the grid, therefore the selection of N and P_{thr} parameters is a trade off between accuracy and computational complexity.

3.3 Comparaison of Standard and DSG Design for Scalar Variable

The standard grid placement is shown in Fig. 3.4 denoted by blue circles. Usage of equidistantly placed grid points for point-mass approximation of the PDF $p_{\text{A}}(x_{k+1}|z^k)$ (3.11) results in the point-mass density $\hat{p}_{\text{A}}(x_{k+1}|z^k;\xi_{k+1})$ (2.6) with the point-mass density approximation error $\tilde{p}_{\text{A}}(x_{k+1}|z^k;\xi_{k+1})$ (3.10), which are plotted in Fig. 3.5 with blue dashed line. The DSG placement is shown in Fig. 3.4, denoted by red crosses and the respective point-mass density $\hat{p}_{\text{A}}(x_{k+1}|z^k;\xi_{k+1})$ (2.6) and the approximation error $\tilde{p}_{\text{A}}(x_{k+1}|z^k;\xi_{k+1})$ (3.10) are shown in Fig. 3.5 with red solid line.

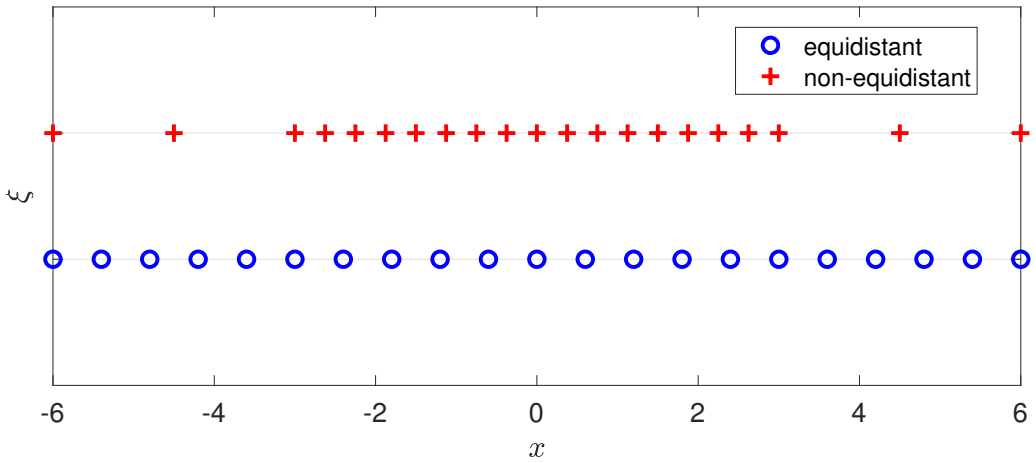

Figure 3.4: Grid points illustration; standard approach with *equal* distances, proposed DSG approach with *non-equal* distances, both $N = 21$.

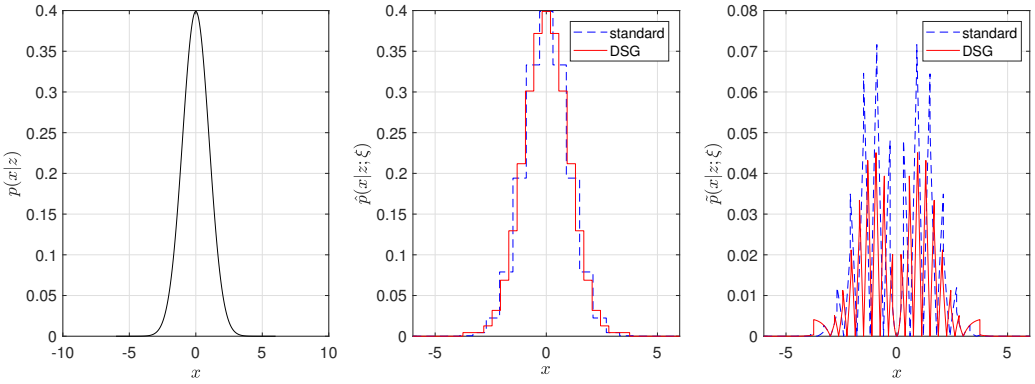

Figure 3.5: PDF, PMG approximation, and PMG approximation error for standard and proposed DSG design.

From the approximation error plot in Fig. 3.5 it can be seen, that the DSG design results in a significantly lower approximation error. To quantify the approximation error of two considered grid design techniques, let the following integral criterion

$$J_{k+1|k} = \int \tilde{p}(x_k|z^m;\xi_k)dx_{k+1}, \tag{3.12}$$

be defined, which can be understood as an overall PDF approximation error. Numerical evaluation of the criterion (3.12) results in

- $J_{k+1|k} = 0.1188$ for standard *equidistant* grid points calculation,

- $J_{k+1|k} = 0.0780$ for proposed *non-equidistant* grid points calculation.

The results indicate that by change from equidistant to non-equidistant grid points location, the point-mass density approximation error can be reduced by more than 30%. To further illustrate the advantage of non-equidistant DSG design a standard Gaussian distribution was approximated by point-mass densities. Three different numbers of grid points N, two splitting ratios defining N_{centre} and N_{tail}, namely $N_{\text{centre}} \approx \frac{2}{3}N$ and $N_{\text{centre}} \approx \frac{3}{4}N$, and two grid design, namely standard and DSG, are considered. The results in the form of the integral criterion $J_{k+1|k}$ (3.12) and relative improvement are shown in Table 3.1. It can be seen, that the proposed non-equidistant DSG design offers a significant improvement in the point-mass density approximation quality compared to the standard equidistant grid design.

N	N_{centre}	N_{tail}	$J_{k+1\|k}$ (std.)	$J_{k+1\|k}$ (**DSG**)	Realative improv. [%]
	2/3	1/3			
21	14	7	0.1188	0.0940	26 %
31	21	10	0.0795	0.0605	24 %
51	35	16	0.0478	0.0356	26 %
	3/4	1/4			
21	17	4	0.1188	0.0780	34 %
31	25	6	0.0795	0.0518	35 %
51	39	12	0.0478	0.0323	32 %

Table 3.1: Quality of point-mass density approximation for standard and density specific grid design.

3.4 Density Specific Grid Design for Vector Variable

The DSG design for the vector state variable follows the idea introduced for the scalar case and can be summarised by the following general algorithm:

Algorithm 4: Density Specific Grid Design (Vector State Variable)

(i) Determination of approximate Gaussian PDF $p_{\text{A}}(\mathbf{x}_{k+1}|\mathbf{z}^k)$ (3.4). Specification of the threshold P_{thr} and number of the grid points N.

(ii) Computation of the region $\mathcal{R}$ in the state space which is to be populated by the grid points. The (hyper-)rectangular region should fulfil the inequality (3.1). A time efficient computation of the region for the vector variable can be based on marginal distributions of $p_{\mathrm{A}}(\mathbf{x}_{k+1}|\mathbf{z}^k)$.

(iii) Split the region $\mathcal{R}$ into two subregions, namely $\mathcal{R}_{\mathrm{centre}}$ and $\mathcal{R}_{\mathrm{tail}}$. The subregion $\mathcal{R}_{\mathrm{centre}}$ is located around the mean value, whereas $\mathcal{R}_{\mathrm{tail}}$ close to the edges of the grid.

(iv) Split N grid points into $N_{\mathrm{centre}} \approx \lceil \frac{3N}{4} \rceil$ and $N_{\mathrm{tail}} = N - N_{\mathrm{centre}}$, where the notation $\lceil \cdot \rceil$ stands for rounding toward positive infinity.

(v) Determine location of N_{centre} grid points $\{\boldsymbol{\xi}_{k+1}^{(i)}\}_{i=1}^{N_{\mathrm{centre}}}$ to equidistantly cover $\mathcal{R}_{\mathrm{centre}}$ and location of N_{tail} grid points $\{\boldsymbol{\xi}_{k+1}^{(i)}\}_{i=N_{\mathrm{centre}}+1}^{N}$ to equidistantly cover $\mathcal{R}_{\mathrm{tail}}$.

The points computed by Algorithm 4 form the *sought* grid for the calculation of the point-mass predictive PDF $\hat{p}(\mathbf{x}_{k+1}|\mathbf{z}^k; \boldsymbol{\xi}_{k+1})$ (2.27) of the PMF.

It should be noticed that the neighbourhood defined by $\boldsymbol{\Delta}_k$ in (2.6) is *not* the same for all grid points anymore. Thus, definition of i-th grid point $\boldsymbol{\xi}_k^{(i)}$ should be always associated with definition of the probability $P_{k|m}(\boldsymbol{\xi}_k^{(i)})$ and *also* with particular $\boldsymbol{\Delta}_k^{(i)}$. The point-mass density has, instead of (2.6), the following form

$$p(\mathbf{x}_k|\mathbf{z}^m) \approx \hat{p}(\mathbf{x}_k|\mathbf{z}^m; \boldsymbol{\xi}_k) \triangleq \sum_{i=1}^{N} P_{k|m}(\boldsymbol{\xi}_k^{(i)}) S\{\mathbf{x}_k; \boldsymbol{\xi}_k^{(i)}, \boldsymbol{\Delta}_k^{(i)}\}. \tag{3.13}$$

3.5 User-Defined Design Parameters and Properties

The proposed basic DSG design is based on two user-defined parameters, namely a splitting ratio for volume of the region $\mathcal{R}$ into two subregions $\mathcal{R}_{\mathrm{centre}}$ and $\mathcal{R}_{\mathrm{tail}}$ and a splitting ratio for number of the grid points N into N_{centre} and N_{tail}. Note that the ratio may (and should) be tuned for particular applications.

If the approximated distribution is not close to symmetric distribution, then the DSG may not result in an improvement in the approximation quality of the PDF by the point-mass density. This is analyzed in Chapter 4, employing various approximation errors, with the presentation of changes in optimal parameters based on approximated PDF. A solution to this problem is proposed in Chapter 5.

3.6 DSG Using UKF Prediction and Filtration

The standard DSG design uses only state dynamics (2.1) and the filtration density $p(\mathbf{x}_k|\mathbf{z}^k)$ known from the measurement update to specify grid support for the predictive density $p(\mathbf{x}_{k+1}|\mathbf{z}^k)$. However, the same grid is also used for definition of the forthcoming filtration density $p(\mathbf{x}_{k+1}|\mathbf{z}^{k+1})$. Moreover, the filtering estimate is usually the one more important for estimation results. Therefore, it seems beneficial to incorporate the information from measurement $\mathbf{z}_{k+1}$ in definition of a grid for both $p(\mathbf{x}_{k+1}|\mathbf{z}^k)$ and $p(\mathbf{x}_{k+1}|\mathbf{z}^{k+1})$.

This is possible by creating another denser part of the grid centred at the approximate filtration estimate $\hat{\mathbf{x}}_{\mathrm{A},k+1|k+1}$(3.2). The size of the grid can be again chosen so it satisfies (3.1) for $p_{\mathrm{A}}(\mathbf{x}_{k+1}|\mathbf{z}^{k+1})$. However, in order to use this grid design the PMF algorithm has to run the UKF prediction, UKF filtration (for grid support region determination) and PMF prediction in predictive step which might not be ideal, mainly for online applications, due to most computational complexity being shifted to one place in the algorithm. This problem is illustrated in Figure 3.8 opposed to usual case shown in Figure 3.7.

Algorithm 5: Density Specific Grid Design using Filtration (Vector State Variable)

(i) Determination of approximate Gaussian PDF $p_{\mathrm{A}}(\mathbf{x}_{k+1}|\mathbf{z}^k)$ (3.4) and $p_{\mathrm{A}}(\mathbf{x}_{k+1}|\mathbf{z}^{k+1})$ similarly. Specification of the threshold P_{thr} (for computation of region $\mathcal{R}$) and smaller $P_{\mathrm{thr,f}}$ (for computation of region $\mathcal{R}_f$), $P_{\mathrm{thr,p}}$ (for computation of region $\mathcal{R}_p$) and allowed number of the grid points N.

(ii) Computation of the region $\mathcal{R}$ (whole grid), $\mathcal{R}_f$ (grid from UKF filtration) and $\mathcal{R}_p$ (grid from UKF prediction), in the state space. The (hyper-)rectangular regions should fulfil the inequality (3.1) for their appropriate probability mentioned in step (i).

(iii) Splitting the number of points N into two regions, one is $\mathcal{R}_{dense} = \mathcal{R}_f \cup \mathcal{R}_p$, the second one is $\mathcal{R}_{sparse} = \mathcal{R} \setminus \mathcal{R}_{dense}$. The points are split so that the $\mathcal{R}_{dense}$ region is populated more densely than region $\mathcal{R}_{sparse}$. See Figure 3.6.

(iv) Determination of location of N_{dense} grid points $\{\boldsymbol{\xi}_{k+1}^{(i)}\}_{i=1}^{N_{\mathrm{dense}}}$ to equidistantly cover $\mathcal{R}_{\mathrm{dense}}$ and location of N_{sparse} grid points $\{\boldsymbol{\xi}_{k+1}^{(i)}\}_{i=N_{\mathrm{dense}}+1}^{N}$ to equidistantly cover $\mathcal{R}_{\mathrm{sparse}}$.

An example of the resulting grid can be seen in Figure 3.6. Insurance of the grid points neighbourhoods not overlapping, as the grid is not trivial, is explained in Chapter 6.

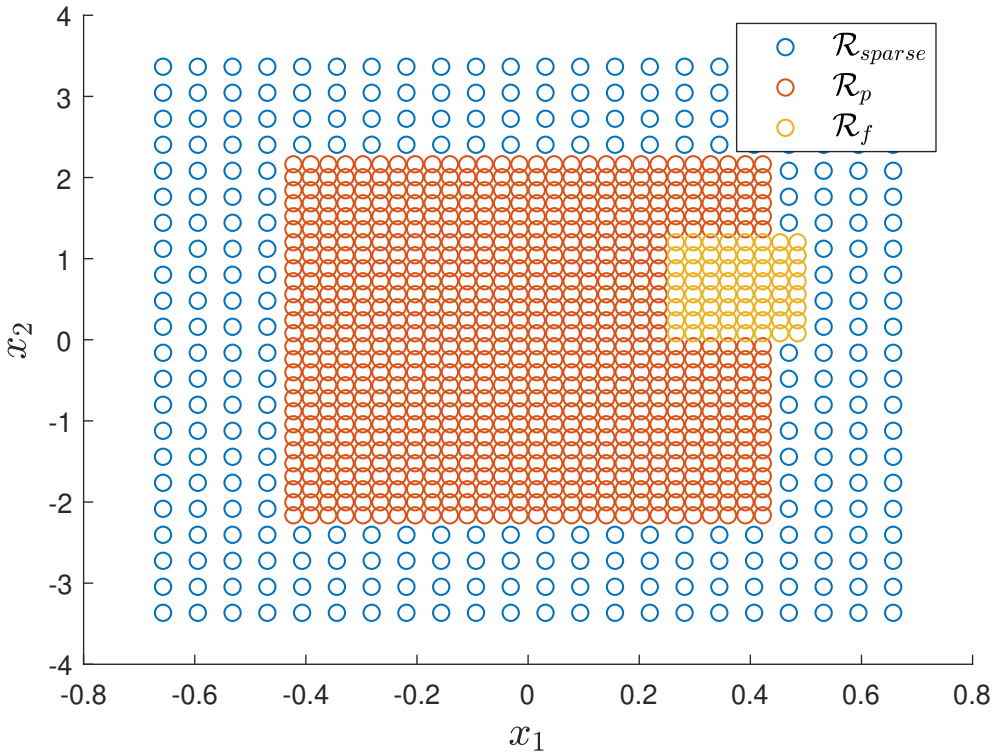

Figure 3.6: Example of grid points distribution in 2 dimensional state space for DSG design with UKF filtration.

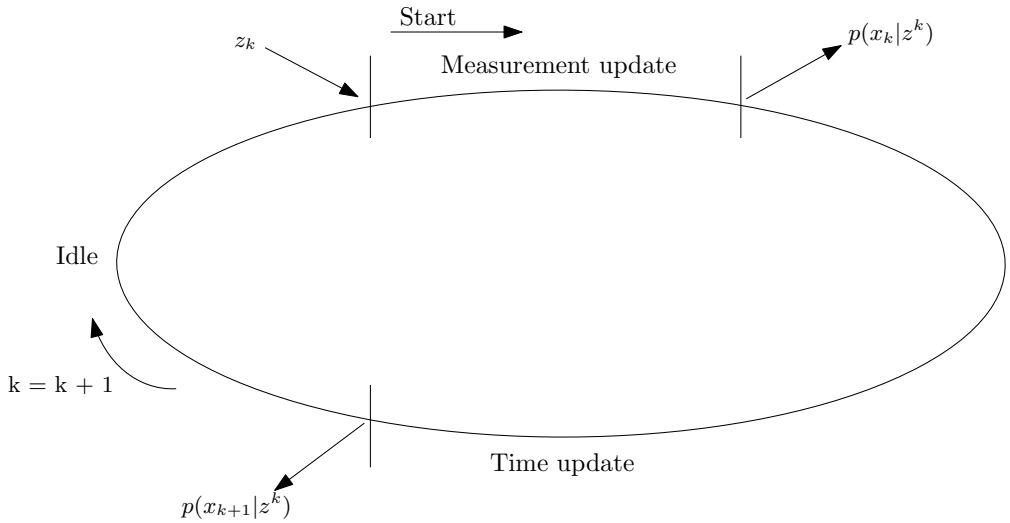

Figure 3.7: Time diagram of most standard filtration algorithms.

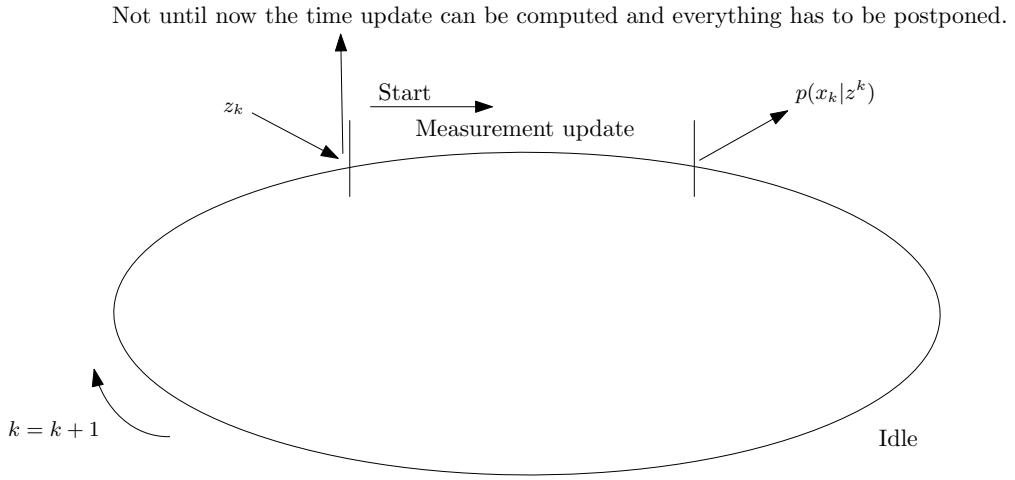

Figure 3.8: Time diagram of PMF using DSGf.

Chapter 4

Analysis of DSG Usage for Non-Gaussian PDF's

The DSG design was proposed and analyzed under an assumption that the approximated PDF is symmetric or is close to being symmetric and also "heavy" around mean value. In this chapter, the influence of higher moments kurtosis/skewness, on the accuracy of the approximation in 1D space is inspected.

Parameters required for the creation of the DSG are the required number of grid points N and the probability that the tail/center grid is supposed to cover (specified as sigma probabilities σ_{out}/σ_{in}). The value N should be chosen as high as possible with the only restriction being the computational capacity i.e. desired time demand. Therefore the parameters that remain to be chosen are σ_{out} and σ_{in}.

There are plenty of criteria that can be used to measure the quality of the approximation. The ones used in this chapter are

- IE - Integral error (7.8) - $\sum_{i=1}^{N_{exc}} \mathsf{abs}\left(p(x=\xi_{exc}^{i};\xi_{exc}) - p(x=\xi_{exc}^{i};\xi_{apx})\right)$,
- ISE - Integral sum error - $1 - \int p(x;\xi_{apx})dx$,
- ME - Mean error - $\mathsf{abs}\left(\mathsf{E}(x) - \hat{\mathbf{x}}_{k|m}(\xi_{apx})\right)$,
- VE - Var error - $\mathsf{abs}\left(\sqrt{\mathsf{var}(x)} - \sqrt{\hat{\mathbf{P}}_{k|m}(\xi_{apx})}\right)$,

where $p(x;\xi_{exc})$ is an "exact" PMD with 30000 points and $p(x;\xi_{apx})$ is the analysed PMD

approximation. The σ combinations are the same for all simulations (each column is corresponding to one index in Figures 8.1 - 8.4)

$$\begin{bmatrix} \sigma_{out} \\ \sigma_{in} \end{bmatrix} = \begin{bmatrix} 8 & 8 & ... & 8 & 7.9 & 7.9 & ... & 1 \\ 7.9 & 7.8 & ... & 0 & 7.8 & 7.7 & ... & 0 \end{bmatrix}. \tag{4.1}$$

The analysis was conducted on gamma distribution Fig. 4.2, Gaussian distribution Fig. 4.1 for comparison and generalized normal distribution Fig. 4.3.

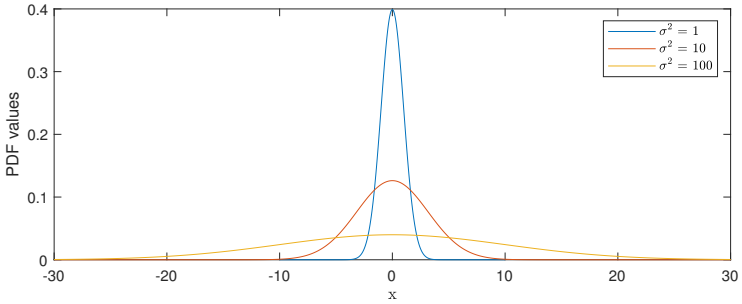

Figure 4.1: Gaussian distributions.

4.1 Gamma Distribution - Skewness and Kurtosis

Gamma distribution was chosen due to the fact that it is specified using only four moments and the analysis can be more intuitive. The approximated PDF

$$\frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-\frac{x}{\theta}}. \tag{4.2}$$

Variance, skewness and kurtosis are given by

$$\sigma^2 = k\theta^2, \tag{4.3}$$

$$\mu^3 = \frac{2}{\sqrt{\theta}}, \tag{4.4}$$

$$\mu^4 = \frac{6}{k} + 3. \tag{4.5}$$

The dynamics of errors based on parameters can be seen in Figures 8.1 - 8.4 in the Appendix. The errors are greater but comparable to ones obtained from approximating Gaussian distribution. Interestingly for the Gaussian distribution, the optimal parameters chosen by IE are independent of the variance of the approximated normal PDF, while other criteria are dependent.

IE criterion is lower for the approximations using DSG grids with outer grid σ between 4 and 5. On the other hand, the rest of errors seem to be smaller for the biggest outer grid and not favouring DSG design that much.

Considering that the estimate, taken only as a mean value, can be "accurate" even though the PDF is not "well approximated" and that PMF's trait is to provide estimate in the form of PMD ergo PDF approximation it might be best to focus on the IE.

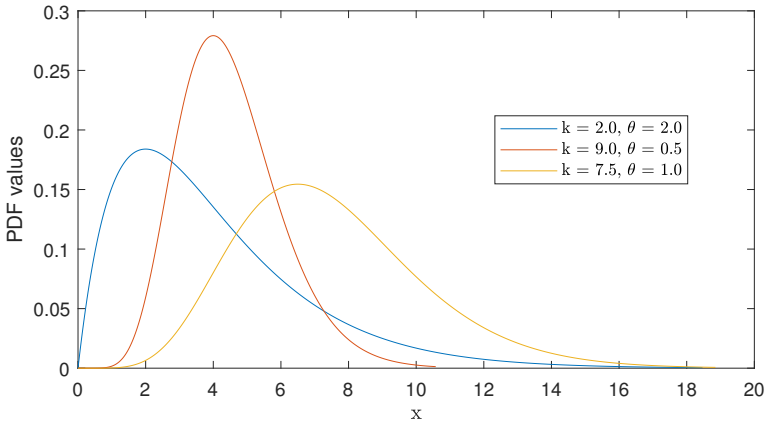

Figure 4.2: Gamma distribution.

4.2 Generalized Normal Distributions - Kurtosis

The generalized normal distribution was also analyzed. For this distribution the kurtosis was varied. However, the findings were the same as for the analysis of gamma distribution.

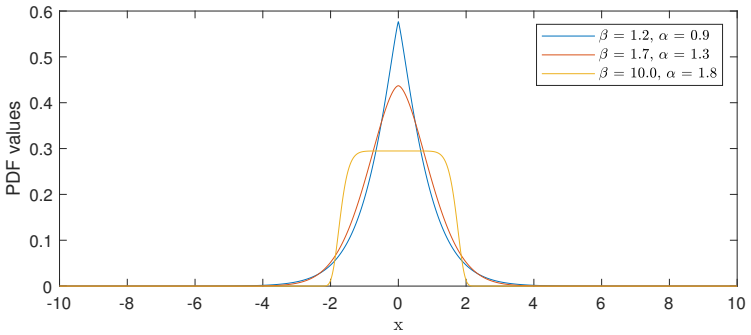

Figure 4.3: Generalized normal distributions.

4.3 Summary

The proposed DSG design has a few shortcomings. The first problem is that for PDFs with dominant higher moments (e.g. skewness, kurtosis) the grid should have more parameters. For example for highly skewed distributions, the denser grid should probably be moved and stretched on one side to compensate for the steepness.

The second problem is that the user has to specify the size of the outer sparse and inner dense grid in the form of the STDs that the grids are supposed to cover. (6.3) To set up those, the user has to possess some insight into the problem. These parameters should ideally be picked automatically by the algorithm.

The third shortcoming is that the user spcified number of STDs are constant in time which might not be ideal as the PDF can change greatly in time. In that case, the ideal choice of those parameters will vary in time. Therefore, ideally, parameters should be time-varying and picked automatically.

Also, for PDF that does not have the "high value"/fast-changing part in the vicinity of mean value, the DSG design is not optimal.

Possible solutions are presented in the following Chapter.

Chapter 5

Density Specific Grid Design Enhancements

In this chapter a solution is proposed to the DSG design shortcomings mentioned in the previous chapter i.e. enhancement is proposed so that compared to the DSG design, the new design utilise full information about the shape of the predictive PMD. The basic idea is summed up in the following steps

1. Design of the sparse (outer) grid and evaluation of the predictive probability at these grid points,
2. Differentiation of “sparse” predictive PMD,
3. Design of the dense (inner) grids in the vicinity of the “sparse” grid points, that are associated with a large numerical derivative.

5.1 Outer Grid

Instead of focusing on moments while setting up the grid, the idea is that, the proposed method exploits a property of the Point-Mass time-update, that a PDF value of any single point can be computed independently. Therefore the algorithm ”probes” the PDF values of carefully picked points in order to choose the size of the grid support.

Algorithm 2: DDG Sparse Grid

---

1. Define a threshold p_{thr} below which the probability at a grid point is considered to be negligible and the grid point does not have to be included into the sparse grid. The threshold setting can be related to computation precision of a given platform.

2. Similarly as in the DSG design [15], compute the predictive moments $\hat{\mathbf{x}}_{\mathrm{A},k+1|k}$ (3.2) and $\mathbf{P}_{\mathrm{A},k+1|k}$ (3.3) and set the number of standard deviations (STDs) n_σ of the marginal distributions of (3.4), which are used for specification of an *a priori* (hyper-)rectangular region $\mathcal{R}_\mathrm{A}$ to be covered by the grid points.

3. At the boundaries of the region $\mathcal{R}_\mathrm{A}$ select a small set of grid points (see example lower) and evaluate the predictive probability at these points. If the probability of the point is (significantly) below the threshold p_{thr}, then the particular point can be shifted towards the mean. On the other hand, if the probability of the point is (significantly) over the threshold p_{thr}, then the particular point can be shifted in an opposite direction from the mean. The shifted points are, then, re-evaluated.

4. When all boundary points are properly set, create a (hyper-)rectangle including all the boundary points, with a smallest possible volume. This (hyper-)rectangle define $\mathcal{R}$. As a consequence, the resulting region $\mathcal{R}$ need not be symmetrical with respect to the mean.

5. Define a cardinality of the sparse grid N_sparse and dense grid $N_\mathrm{dense} = N - N_\mathrm{sparse}$. Fill the region $\mathcal{R}$ by equidistantly placed N_sparse hyper-rectangular grid points $\{\boldsymbol{\xi}^{(j)}_{\mathrm{sparse},k+1}\}_{j=1}^{N_\mathrm{sparse}}$ (each sparse grid point is associated with a neighbourhood $\boldsymbol{\Delta}_{\mathrm{sparse},k+1}$).

The illustration for two-dimensional state and points in the rectangles corners and middles of the sides can be seen in Figure 5.1, where $\mathbf{m}$ is the UKF prediction. The rectangular shape is chosen to keep the computational complexity low.

Note that this method might not yield the best results if the approximated PDF is multimodal and the areas between modes have the negligible value of PDF. However, this problem might be lessened by forcing the points to have negligible probability twice in a row before keeping them set in place.

Numerical illustration for these enhancements is present in Chapter 7.

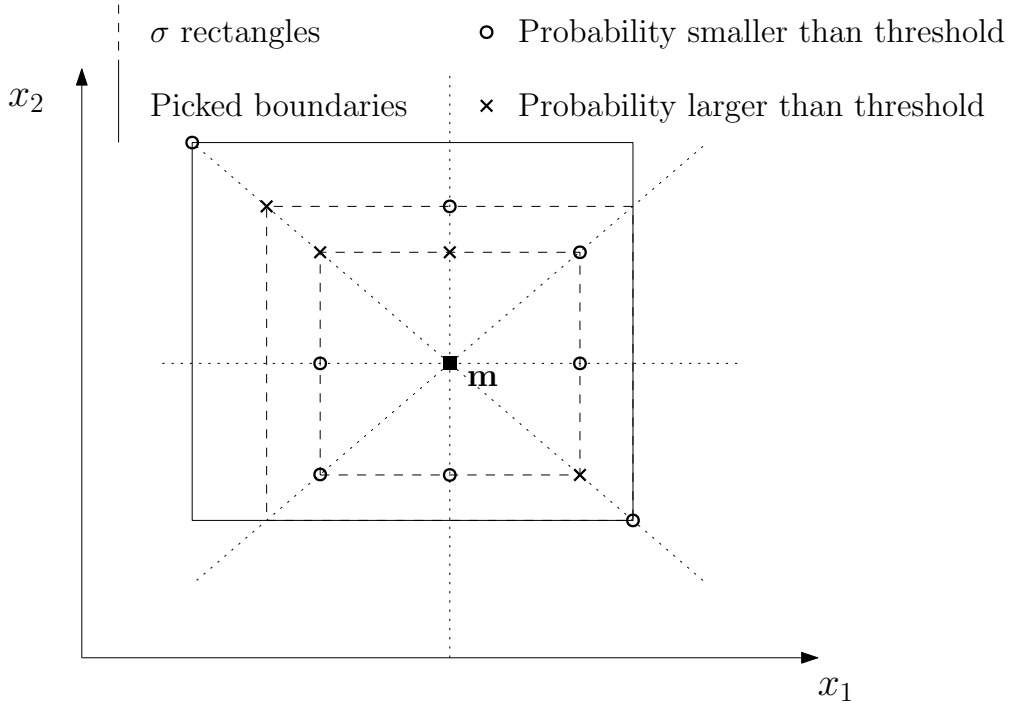

Figure 5.1: Illustration of search for outer grid boundaries.

5.2 Inner Grid

This section offers an alternative way of setting up the denser parts. Again the focus is steered away from the moments of the approximated PDF to the ability to compute the probability of any single point. The process is as follows.

Algorithm 3: Sparse Grid PMD Differentiation

---

1. Compute the PMD for N_{sparse} sparse grid points according to the convolution (2.28).
2. For each sparse grid point $\boldsymbol{\xi}^{(j)}_{\text{sparse},k+1}$ compute the divided difference vector $\mathbf{d}^{(j)}_{k+1} \in \mathbb{R}^F$

according to

$$\mathbf{d}_{k+1}^{(j)} = \left[\frac{\mathsf{abs}(P_{k+1|k}(\boldsymbol{\xi}_{\text{sparse},k+1}^{(j)}) - P_{k+1|k}(\boldsymbol{\xi}_{\text{sparse},k+1}^{(\imath_1)}))}{\left\| \boldsymbol{\xi}_{\text{sparse},k+1}^{(j)} - \boldsymbol{\xi}_{\text{sparse},k+1}^{(\imath_1)} \right\|_2}, \ldots, \right. \tag{5.1}$$

$$\left. \frac{\mathsf{abs}(P_{k+1|k}(\boldsymbol{\xi}_{\text{sparse},k+1}^{(j)}) - P_{k+1|k}(\boldsymbol{\xi}_{\text{sparse},k+1}^{(\imath_F)}))}{\left\| \boldsymbol{\xi}_{\text{sparse},k+1}^{(j)} - \boldsymbol{\xi}_{\text{sparse},k+1}^{(\imath_F)} \right\|_2} \right], \tag{5.2}$$

where the index vector $\imath = [\imath_1, \imath_2, \ldots, \imath_F]$ is selected so that the closest neighbouring points are chosen and $\|\|_2$ is the Euclidean distance. The F depends on what points are chosen as adjacent, this can be tailored to particular case. If the point is on the boundary of the grid, the difference is counted only to the existing grid points.

3. Compute the grid points rating according to

$$d_{k+1}^{(j)} = \sum_{i=1}^{F} \mathbf{d}_{k+1}^{(j)}(i), \tag{5.3}$$

which characterises the overall variability of the PDF in the vicinity of the j-th sparse grid point $\boldsymbol{\xi}_{\text{sparse},k+1}^{(j)}$.

Illustration of this algorithm is in Figure 5.2.

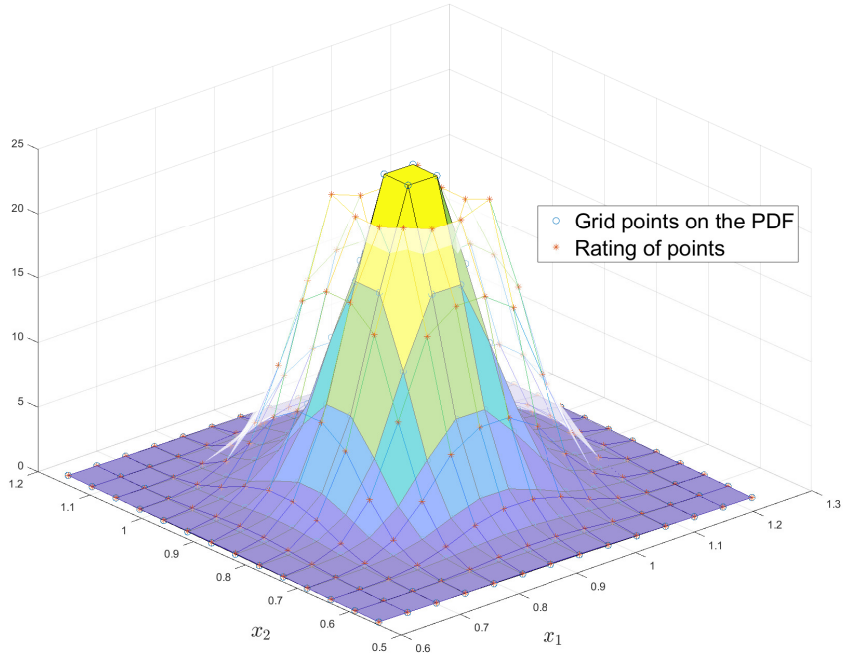

Figure 5.2: Example of points ratings.

Adjacent points for this thesis implementation are picked as shown in Figure 5.3. For faster computation in more dimensional state the diagonal grid points do not have to be taken as adjacent.

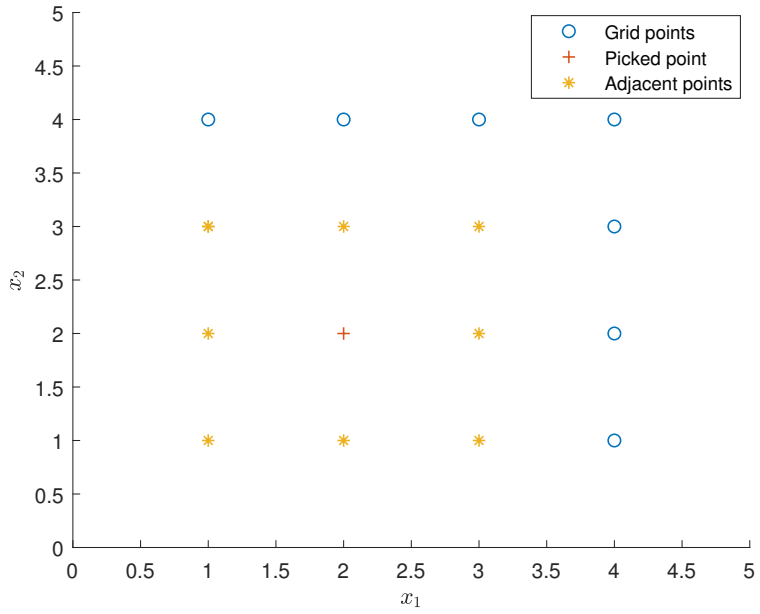

Figure 5.3: Illustration of adjacent points.

Having defined sparse grid points (Algorithm 2) and computed their difference rating (Algorithm 3), which in fact precisely denotes the state-space regions with significant change or variability of the conditional PDF, it is possible to construct the dense grid according to the following algorithm:

Algorithm 4: Dense Grid Design

---

1. Sort the sparse grid points $\{\boldsymbol{\xi}^{(j)}_{\text{sparse},k+1}\}_{j=1}^{N_{\text{sparse}}}$ in a descending order according to their rating $d^{(j)}_{k+1}$ (5.3).

2. Set the splitting ratio r, which determines number of dense grid points r^{n_x} used for covering the vicinity of the j-th sparse grid $\boldsymbol{\xi}^{(j)}_{\text{sparse},k+1}$. Reasonable choice is $r \in \{2, 3\}$[1], but note that the ratio can be varying and it can depend on the rating $d^{(j)}_{k+1}$ (5.3); the higher rating, the higher ratio.

3. Recursively split the vicinity of the ordered grid points, starting from the points with high rating $d^{(j)}_{k+1}$, until the number of the dense grid points N_{dense} is reached.

[1] As the DDG design can effectively and precisely find the denser grid design regions, the splitting ration can be 3 as opposed to the DSG design splitting ratio 2.

In Figure 5.4 the created grid support, for a PDF close to Gaussian, can be seen.

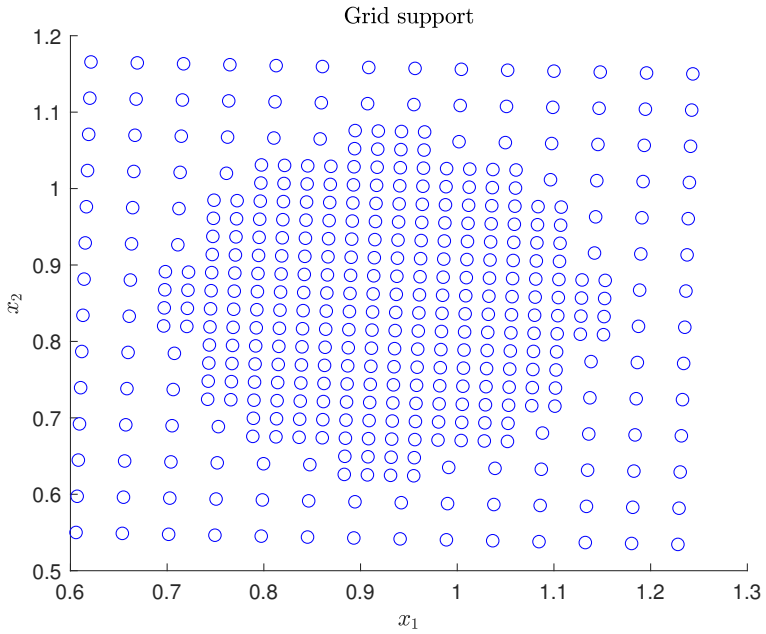

Figure 5.4: Example of grid support.

5.3 Summary

With these enhancements, the user parameters changed from setting up two σ probabilities and the desired number of grid points N, to picking just N and the ratio between split and non-split points. This effectively solves any problems with the presence of higher moments in the approximated PDF as long as discussion in Section 5.1 is kept in mind. It also enables the grid to better adapt in time.

It can be seen that, compared to the recently proposed DSG, the DDG dense (inner) grid need not be rectangular and thus it more realistically covers the support of the conditional PDF with the significant volume without any requirement on computation and utilisation of a set of higher-order moments.

Chapter 6

Implementation in NEF Toolbox

The NEF is a publicly available[1] Matlab toolbox designed for a non-linear state estimation and identification of discrete dynamic stochastic systems described by the state-space model of the form (2.1), (2.2). The toolbox consists of mutually linked classes which implement all functionality needed for state estimation problem. The toolbox has implemented classes for

- System description,
- Random variable description,
- Function definition,
- Estimator definition,
- Performance evaluation,
- System identification.

The *System description* allows a specification of the system model in a probabilistic or a structural fashion. The *Random variable description* (RV) enables a description of the characteristics of RVs related to the model or the estimate. The following PDFs are supported: uniform, beta, gamma, Gaussian, Gaussian sum, empirical and now also point-mass. The *Function definition* enables a description of equations in the state and measurement equations or PDFs of RVs in the case of the probabilistic system description. The *Estimator*

[1]The toolbox is available at `http://ntf.kky.zcu.cz/nef`.

definition contains various local and global estimation algorithms as the Kalman filter (KF), extended KF, unscented KF, divided difference filters, particle filter, Gaussian sum filter, the ensemble KF, and newly the PMF. The part of NEF for *Performance evaluation* enables calculation of various performance metrics related to the estimate quality (mean square error (MSE), non-credibility index (NCI), etc.). The category *System identification* contains methods for the estimation of parameters of a linear state-space model based on measured data. The NEF thus represents a powerful and complex tool for the state and parameter estimation of linear and nonlinear systems. Further information on the toolbox structure can be found in [12].

To add the PMF with standard and newly proposed DSG design, as a powerful filter suitable for highly nonlinear and non-Gaussian systems, into the NEF estimator repertoire, two subclasses were implemented; first, for random variable description, allowing manipulations with a point-mass density, second, for estimator definition defining the point-mass estimator. Both subclasses are particularized below.

Note that due to NEFs firmly given structure and philosophy, the DSG design using UKF filtration to specify the second denser part of the grid 3.6, was not implemented.

6.1 Random Variable Subclass: nefPointMassRV

First newly implemented class `nefPointMassRV` is a subclass of the parent random variable class `nefRV`. The implemented subclass has been designed to easily *create and describe* point-mass density $p(\mathbf{x}_k|\mathbf{z}^m)$ (2.6), *compute moments*, and *generate samples*. It can also plot PDFs for one and two dimensional random variables.

The subclass supports a description of point-mass densities that are constant on an equally and unequally spaced neighbourhood. An instance of the class `nefPointMassRV` with *equidistant* grid can be created (in Matlab) as follows

```
PointMassDensity = nefPointMassRV(gridPoints, pointsProb,
Delta, 'gridType', 'eqdist');
```

where the input parameters are

- `gridPoints` - a matrix of dimension $\mathbb{R}^{n_x \times N}$ containing all available grid points $\{\boldsymbol{\xi}_k^{(i)}\}_{i=1}^N$ so that i-th column of the matrix `gridPoints` is the i-th point $\boldsymbol{\xi}_k^{(i)}$,

- `pointsProb` - a vector of dimension $\mathbb{R}^{N\times 1}$ containing probabilities $P_{k|m}(\boldsymbol{\xi}_k^{(i)}), \forall i$,
- `Delta` - a vector of dimension $\mathbb{R}^{n_x\times 1}$ defining a neighbourhood of a grid point herein denoted as $\boldsymbol{\Delta}_k$,
- `'gridType', 'eqdist'` - a name-value pair meaning that the created grid will have equidistantly distributed points.

The N is the total number of grid points. An instance of the class `nefPointMassRV` with *non-equidistant* grid, which is needed for the DSG design, is created analogously to the previous instance as

```
PointMassDensity = nefPointMassRV(gridPoints, pointsProb, Delta,
'gridType', 'noneqdist');
```

with two differences

- `Delta` is a matrix of dimension $n_x \times N$ (same dimension as `gridPoints`), where a neighbourhood of i-th grid point, i.e., $\boldsymbol{\Delta}_k^{(i)}$ in (3.13), is i-th column of the matrix,
- a value `'noneqdist'` is used in place of `'eqdist'`.

6.2 Estimator Subclass: nefPMF

The second newly implemented class `nefPMF` is a subclass of the parent estimator class `nefEstimator`. The implemented subclass has been designed to easily *create* point-mass estimator defined by Algorithm 1 and *estimate* the state. As any of the other estimators implemented in the NEF Toolbox [12], the point-mass estimator can be configured to perform filtering, prediction and smoothing task.

A basic instance of the class `nefPMF` can be created (in Matlab) as follows

```
pmfEst = nefPMF(system, 'gridSigma', gridS, 'noGridPoints', N);
```

where

- `system` is an instance of the NEF class `nefSystem` containing state-space model (2.1), (2.2). Definition of the model in the NEF Toolbox is illustrated in [12],

- `'gridSigma', gridS` is a name-value pair specifying σ-probability that the grid is trying to cover in predictive step (Fig. 6.1),
- `'noGridPoints', N` is a name-value pair specifying the number of grid points N to be used in predictive grid design.

For use of DSG (section 3.2) design name-value pair `'gridSigmaPredIn', gridSp`, specifying σ-probability that the inner dense grid is going to cover in predictive step, has to be added (Fig. 7.4).

The instance `nefPMF` provides estimates for all required time steps in the form of instances of density class `nefPointMassRV`. Note that the class `nefPointMassRV` includes methods for moment computations.

6.3 Grid Design in NEF

Particular application of PMF time-update grid creation in NEF can be summarized by the following steps (note that, while the algorithm is general, the accompanying figures are all for two dimensional state) :

Algorithm 6: NEF PMF time-update grid creation

(i) Approximate predictive moments $\hat{\mathbf{x}}_{\mathrm{A},k+1|k}(3.2), \mathbf{P}_{\mathrm{A},k+1|k}(3.3)$ are counted using UKF predictive step.

(ii) Grid boundaries are found as the (hyper-)rectangle around $l\sigma-$ellipse of $\mathcal{N}\{\mathbf{0}, \mathsf{diag}(\mathbf{P}_{\mathrm{A},k+1|k})\}$, where l is a parameter chosen by the user during estimator class instance creation. This can be seen in Figure 6.1, though note that in the figure the boundaries are already rotated and moved to the $\hat{\mathbf{x}}_{\mathrm{A},k+1|k}$. The $k\sigma-$ellipse was discussed in 3.1.1.

(iii) Area inside grid boundaries is filled with evenly distributed points so that the number of points for each dimension is the same, for example the grid is $2 \times 2, 15 \times 15, 3 \times 3 \times 3$ etc.

(iv) If the user specifies non zero $k\sigma$ probability for the inner denser grid, then the new denser grid boundaries are created analogically to the step ii. All points belonging inside the boundaries are then split into 2^{n_x} points, where n_x is the dimension of the state, in a way that the neighborhoods $\mathbf{\Delta}_k$, of the outer sparse grid, stay the same. This step, for $n_x = 2$, is shown in Figure 6.3.

(v) The grid is moved to $\hat{\mathbf{x}}_{\mathrm{A},k+1|k}$ and rotated so the grid boundaries are parallel with $\mathbf{P}_{\mathrm{A},k+1|k}$ eigenvectors. The possible outcome can be seen in Figure 6.1. The rotation does not change shape of the neighbourhoods $\mathbf{\Delta}_k$ of points as represented in Figure 6.2.

As follows from (iii), if the n_x-th root of the number of sought grid points is not a whole number, the total number of points has to be bigger/smaller. NEF implementation rounds the number of points down.

Note that the number of points N_s in a sparse grid created in (iii), when using DSG design, has to be set up in a way, so that the total number of points after point division is (nearly) equal to the demanded number of points. The N_s is estimated as ratio between area of denser and more sparse grid as

$$N = \frac{N}{2^{n_x} \cdot r - r + 1}, r = \frac{\prod \mathbf{b}_{in}}{\prod \mathbf{b}_{out}} \tag{6.1}$$

where n_x is state dimension, $\mathbf{b}_{in}$ and $\mathbf{b}_{out}$ are coordinates of such a point that is in the most distant corner from origin of the denser respectively sparser grid area bounding (hyper-)rectangle, before its rotation and shifting to $\hat{\mathbf{x}}_{\mathrm{A},k+1|k}$ and $\prod$ represents product of all elements of the consequent vector. Example of how to use NEF for PMF estimation is shown in Subsection 7.2.1.

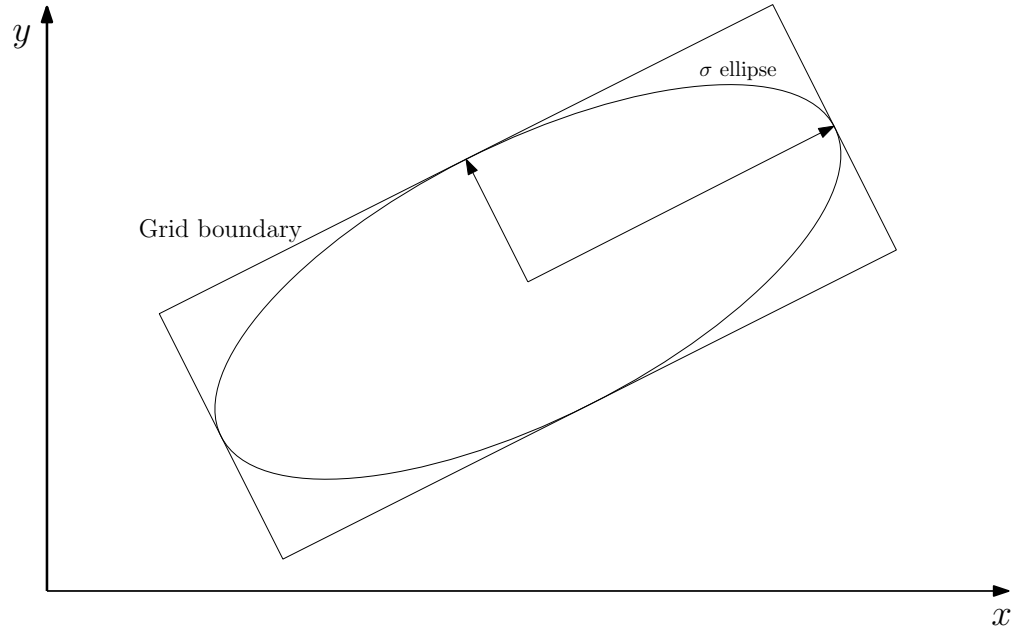

Figure 6.1: Illustration of grid boundaries created from σ ellipse in time update step.

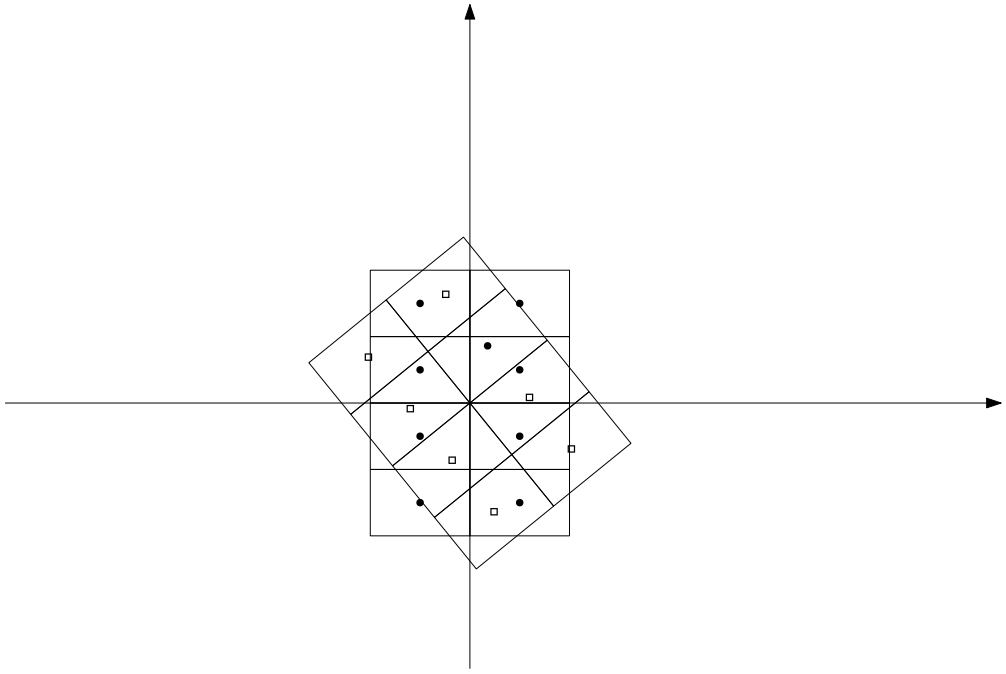

Figure 6.2: Illustration of rotation of a grid by eigenvectors.

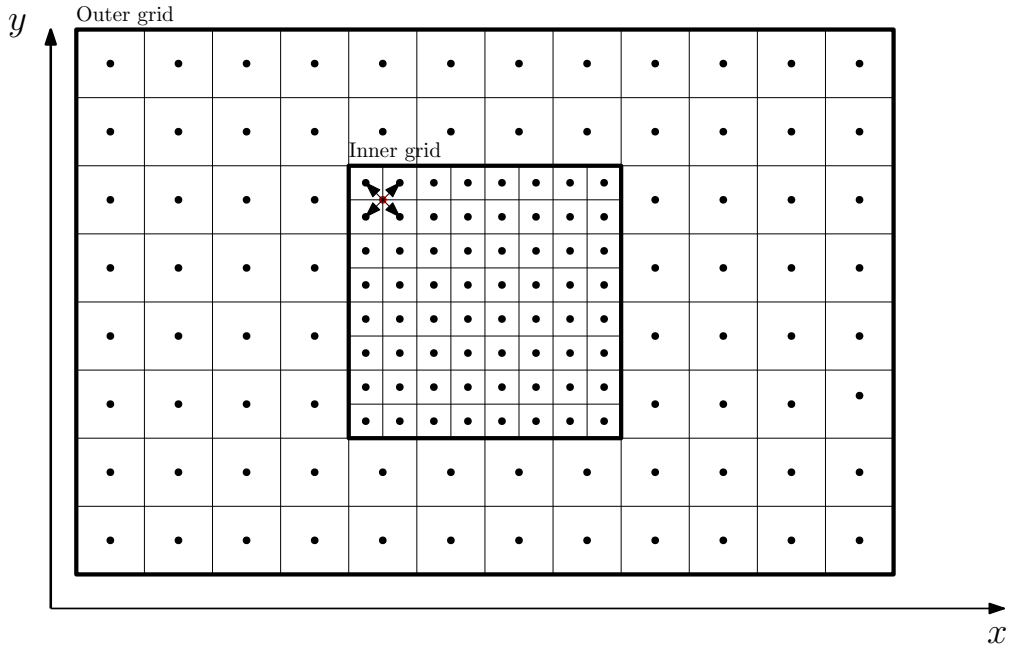

Figure 6.3: Illustration of grid points distribution in 2 dimensional state space for DSG design.

Chapter 7

Simulations and Results

Firstly, a static approximation case is shown in order to demonstrate the DSG and DDG differences and performance. Secondly, this chapter presents simulation results for the predator-prey model from the NEF toolbox for all of the hereinbefore presented algorithms. After that, results for terrain-aided navigation scenarios implemented outside NEF are shown.

7.1 Static Case

A Gaussian mixture was approximated by DSG and DDG designs. This mixture consisted of four components $\mathcal{N}\{\mathbf{x}; \hat{\mathbf{x}}_i, \mathbf{P}_i\}$ with weights w_i, where

$$\mathbf{P}_1 = \mathbf{P}_2 = \mathbf{P}_3 = \mathbf{P}_4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{7.1}$$

$$\left[\begin{array}{c|c|c|c} \hat{\mathbf{x}}_1 & \hat{\mathbf{x}}_2 & \hat{\mathbf{x}}_3 & \hat{\mathbf{x}}_4 \end{array} \right] = \begin{bmatrix} 1 & 2 & 4 & 5 \\ 4 & 1 & 3.5 & -3.5 \end{bmatrix}, \tag{7.2}$$

$$\left[\begin{array}{cccc} w_1 & w_2 & w_3 & w_4 \end{array} \right] = \begin{bmatrix} 0.1 & 0.3 & 0.4 & 0.2 \end{bmatrix}. \tag{7.3}$$

The PDF can be seen in Figure 7.1. The errors while approximating this mixture by Standard, DSG and DDG are presented in Tables 7.1, 7.2 and 7.3. Important to note is that the $\sigma_{out} = 2$ and $\sigma_{in} = 1$ were set up after experiments so the approximation yields best results. Integral and variance error are smillar, while mean and integral sum error are much lower when using DDG. Inconsistency when using DSG for approximation of inappropriate distributions can be also noted.

The Figure 7.2 shows the creation of the outer grid boundaries for this case. The boundaries are similar to the ones gotten when using DSG except for asymmetry which is introduced by the highlighted "probe" point, see Chapter 5. In Figure 7.3 the used grids can be seen.

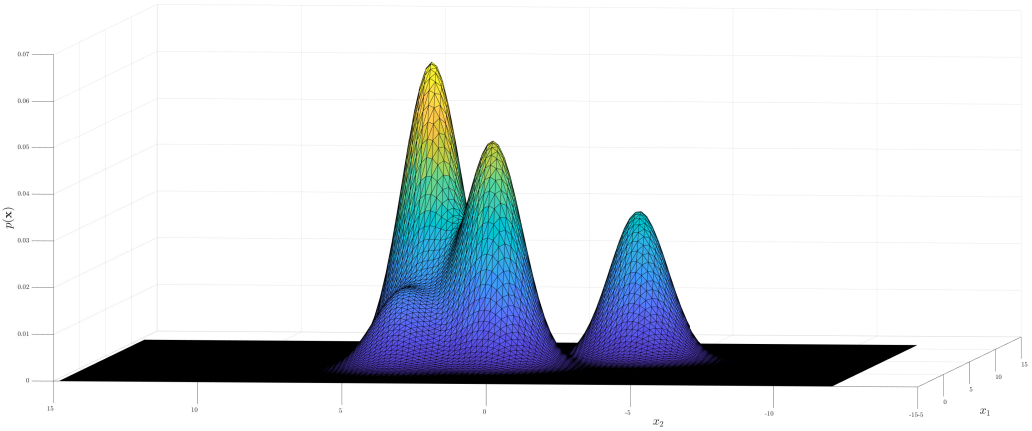

Figure 7.1: Gaussian mixture.

DDG	50	200	500
ISE	$1.1102 \cdot 10^{-15}$	$2.2204 \cdot 10^{-16}$	$2.2204 \cdot 10^{-16}$
ME	0.2439	$2.5314 \cdot 10^{-5}$	$6.7043 \cdot 10^{-6}$

Table 7.1: Approximation errors DDG design.

DSG	50	200	500
ISE	0.0733	$6.3488 \cdot 10^{-5}$	$3.8642 \cdot 10^{-5}$
ME	0.7567	$6.1405 \cdot 10^{-5}$	$3.1102 \cdot 10^{-4}$

Table 7.2: Approximation errors DSG design.

EQ	50	200	500
ISE	0.0468	0.0062	$5.5585 \cdot 10^{-8}$
ME	1.5848	0.0606	$2.4479 \cdot 10^{-5}$

Table 7.3: Approximation errors equal grid design.

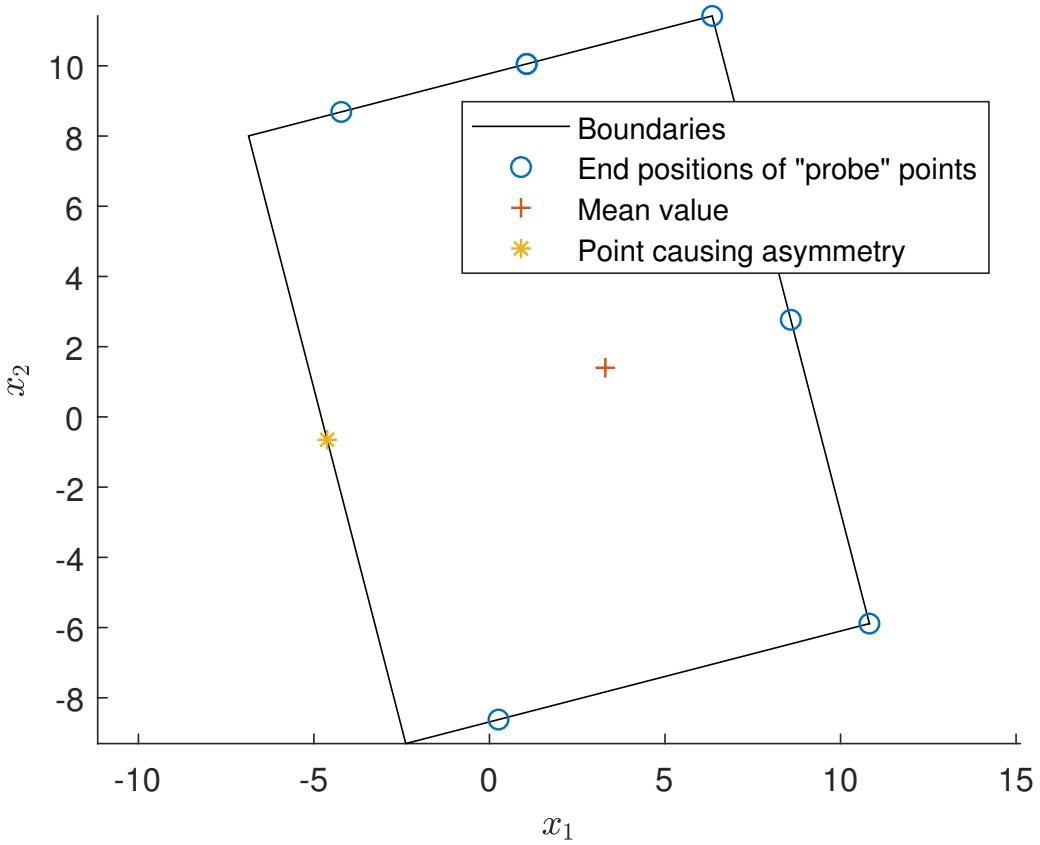

Figure 7.2: Outer grid boundaries creation.

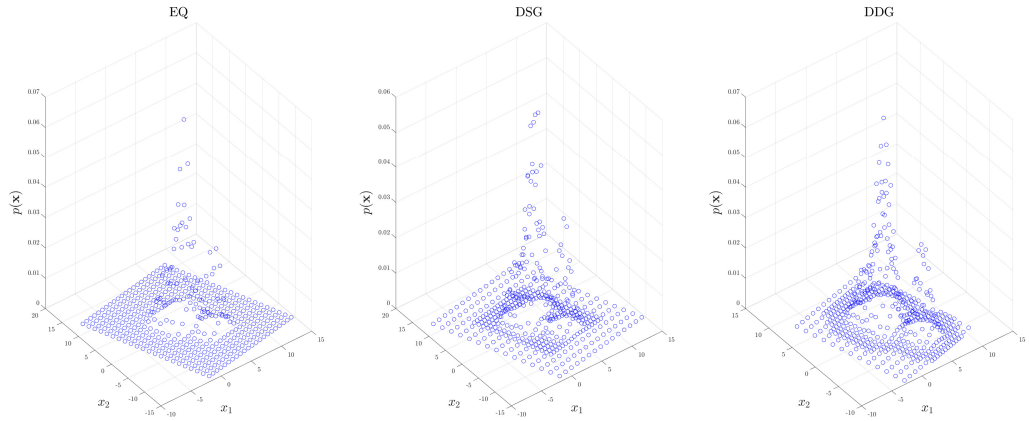

Figure 7.3: Example of grids for each filter.

7.2 Dynamic State Estimation using NEF Toolbox

In this section, simulation results from NEF toolbox are presented.

7.2.1 Predator-Prey Model - PMF Standard and PMF DSG

The PMF was implemented for the following two-dimensional nonlinear predator-prey model

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{x}_{k+1}(1) \\ \mathbf{x}_{k+1}(2) \end{bmatrix} = \begin{bmatrix} T \cdot (1-a)\mathbf{x}_k(1) + Tb\mathbf{x}_k(1)\mathbf{x}_k(2) \\ T \cdot (1+c)\mathbf{x}_k(2) + Td\mathbf{x}_k(1)\mathbf{x}_k(2) \end{bmatrix} + \mathbf{w}_k, \tag{7.4}$$

$$z_k = \mathbf{x}_k(1) + v_k, \tag{7.5}$$

where the constants are $T = 1$, $a = 0.04$, $b = c = d = 0.08$ and $\mathbf{x}(j)$ means j-th element of the vector $\mathbf{x}$. This model belongs to test examples in NEF Toolbox. The state and measurement noises are described by the zero-mean Gaussian PDFs with covariance matrices

$$\mathbf{Q} = \begin{bmatrix} 0.005 & 0 \\ 0 & 0.005 \end{bmatrix}, \; R = 0.01, \tag{7.6}$$

respectively. The state initial condition is also a Gaussian random variable with PDF

$$p(\mathbf{x}_0) = \mathcal{N}\{\mathbf{x}_0; \begin{bmatrix} 0.9 \\ -0.85 \end{bmatrix}, \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}\}. \tag{7.7}$$

NEF Toolbox: Model Definition and Trajectory Simulation

The model (7.4)–(7.7) can be easily defined by the following set of commands [12]:

```
% number time steps
K = 10;

% function f in state equation
f = nefHandleFunction(@(x,u,w,k)[0.96*x(1)+0.08*x(1)*x(2)+w(1);
1.08*x(2)-0.08*x(1)*x(2)+w(2)],[2 0 2 0])
% function h in measurement equation
H = [1 0];
h = nefLinFunction(H,[],1);

% state noise
```

```
Q = eye(2)*0.005;
w = nefGaussianRV([0 0]',Q);
% measurement noise
R = 0.01;
v = nefGaussianRV(0,R);

% initial condiditon
x0 = nefGaussianRV([0.9;-0.85],1e-3*eye(2));

% creating system
system=nefEqSystem(f,h,w,v,x0);
```

The model trajectory can be simulated using:

```
[z,x] = simulate(system,K,[]);
```

NEF Toolbox: PMF Design and State Estimation

A set of three PMF is configured, namely the following instances of the class `nefPMF` are created:

- `PMF_exact` with $N = 1000$ equidistantly placed grid points; this filter is assumed to produce "almost" true state estimates,
- `PMF_standard` with $N = 112$ *equidistantly* placed grid points,
- `PMF_DSG` with $N = 112$ *non-equidistantly* placed grid points.

The above mentioned instances can be defined by the following commands:

```
PMF_exact = nefPMF(system,'gridSigma',6,'noGridPoints',961);
PMF_standard = nefPMF(system,'gridSigma',6,'noGridPoints',100);
PMF_DSG = nefPMF(system,'gridSigma',6,'noGridPoints',112,
'gridSigmaPredIn',4);
```

Having the generated data and PMFs' instances, the state estimation can be realised by

```
[est_PMFexact] = estimate(PMF_exact,z,[]);
[est_PMFstandard] = estimate(PMF_standard,z,[]);
[est_PMFdsg] = estimate(PMF_DSG,z,[]);
```

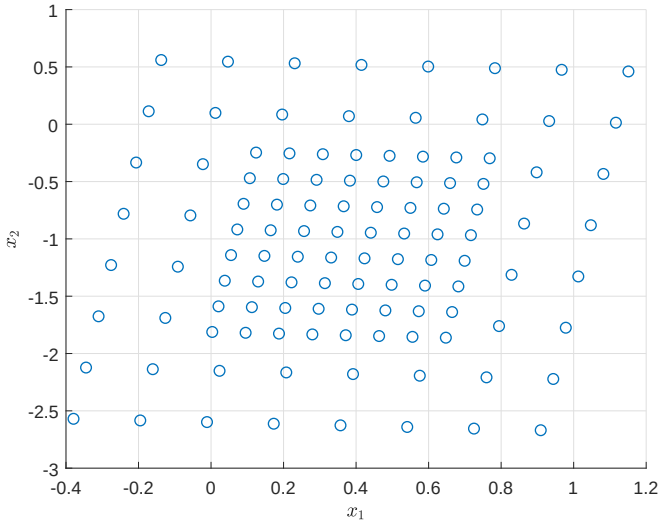

Figure 7.4: Example of grid at step $k = 10$ for `PMF_DSG`.

Example of points distribution for DSG design can be seen in Fig. 7.4. The final estimates are as the instances of the class `nefPointMassRV`.

NEF Toolbox: Results

The state estimates mean square errors[1] (MSE) and time complexity averages from 1000 Monte-Carlo simulations are summarized in Table 7.4. An example of a grid points distribution at step $k = 10$ for the filter `PMF_DSG` can be seen in Fig. 7.4. In Fig. 7.5 there are examples of PDFs for each filter in the same step. It is apparent that the `PMF_DSG` is achieving a better approximation of the part of the PDF with high probability, that mostly contributes to the computation of moments, by using more points than the `PMF_standard` filter.

Before analysing the results in Table 7.4 it is worth noting, that the estimate performance in this table is compared using the mean square error criterion (MSE) in addition to the criterion $J_{k+1|k}$ (3.12) to illustrate the impact of the approximation quality of the point-mass density on the PMF mean estimate. The criterion $J_{11|10}$ (3.12) is, in this case with unknown true density, computed as a difference between the point-mass density of the `PMF_exact` (assumed to be almost true) and point-mass density of either the `PMF_standard` or the

[1]Moments of any distribution in the NEF Toolbox can be easily calculated as well as various estimate performance criteria such as MSE, NCI, etc. Details can be found in [12] and references therein or on the toolbox homepage.

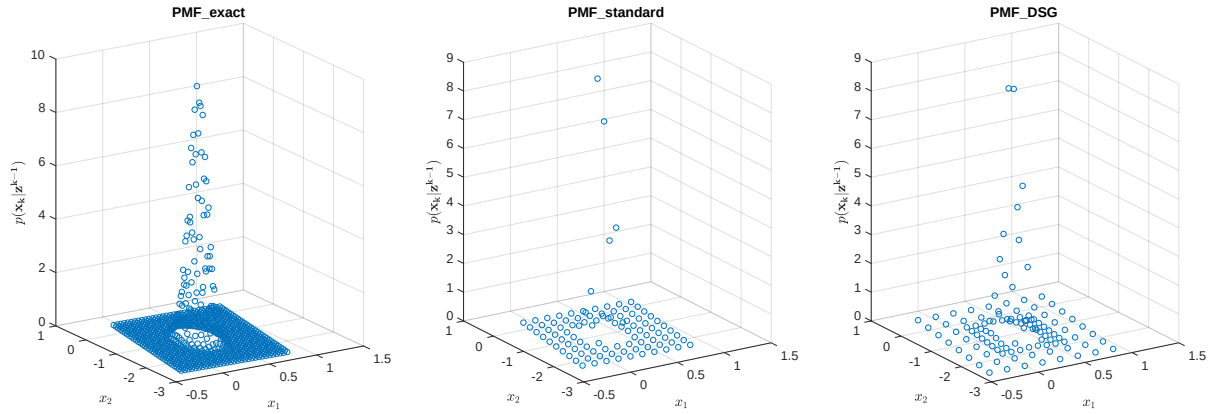

Figure 7.5: Measurement update pdf for each filter in step $k = 10$.

	MSE	MSE Relative Deterioration [%]	$J_{11\|10}$	Time [sec]
`PMF_exact`	0.01546	–	–	0.833
`PMF_standard`	0.01618	5	0.011	0.045
`PMF_DSG`	0.01548	0.1	0.005	0.048

Table 7.4: Performance of particular PMF designs.

`PMF_DSG`, i.e., as

$$\tilde{p}(\mathbf{x}_{k+1}|\mathbf{z}^k;\boldsymbol{\xi}_{k+1}) = |\hat{p}_{\text{exact}}(\mathbf{x}_{k+1}|\mathbf{z}^k;\boldsymbol{\xi}_{k+1}) - \hat{p}(\mathbf{x}_{k+1}|\mathbf{z}^k;\boldsymbol{\xi}_{k+1})|, \tag{7.8}$$

where $\hat{p}_{\text{exact}}(\mathbf{x}_{k+1}|\mathbf{z}^k;\boldsymbol{\xi}_k)$ is a density of the `PMF_exact` and $\hat{p}(\mathbf{x}_{k+1}|\mathbf{z}^k;\boldsymbol{\xi}_{k+1})$ is a density of the `PMF_standard` or the `PMF_DSG`. Table 7.4 also shows the relative deterioration of the MSE estimation performance of the `PMF_standard` and the proposed `PMF_DSG` with respect to `PMF_exact` and the average estimation time for all time steps.

The table reveals that the proposed filter `PMF_DSG` provides significantly more accurate not only the density estimates but also the mean estimates using the same number of grid points as the filter `PMF_standard`, therefore preserving the computational complexity.

7.2.2 Predator-Prey Model - PMF DSG and Other Filters

For the same model 150 Monte-Carlo (MC) simulations with K = 20 were done in order to compare PMF DSG design against other filters, mainly against PF. The filters were setup by the following commands

```
estimators = {...
nefUKF(model),... % standard UKF
nefSUKF(model),...% square-root version of the UFK
nefKalman(model),...% Extended Kalman filter in Josephs form
nefSKalman(model),...% squre-root variant of the Extended Kalman filter
nefPMF(model,'gridSigma',6,'noGridPoints',400,'gridSigmaPredIn',3),...% DSG
    Point mass filter
nefPMF(model,'gridSigma',6,'noGridPoints',400),...% Points mass filter
nefPF(model_pdf,'sampleSize',180)% Particle filter
};
```

so that PF and PMF takes approximately the same time to compute estimates for one MC simulation. The Table 7.5 shows comparaison between standard PMF, DSG PMF, PMF auto with automatically set up boundaries 5.1 and PMF DDG using the difference method for denser grid creation (see Section 5.2). RMSE is a root of the MSE. It is shown that PMF filter can be setup in a way that it is more accurate than other filters and also more accurate than PF with similar computational time. The PMF auto is more precise than standard PMF. Also, even thought the approximated PDF is close to a Gaussian PDF, the PMF DDG performs relatively well compared to the PMF DSG.

Filter	Predator RMSE	Prey RMSE	Time
UKF	0.23113	0.069587	0.021447
sUKF	0.23113	0.069587	0.046944
EKF	0.23117	0.069592	0.019402
sEKF	0.23117	0.069592	0.044937
PMF	0.23129	0.069606	0.13489
PMF auto	0.23075	0.069588	0.12917
PMF DDG	0.23067	0.06959	0.19801
PMF DSG	0.23067	0.069571	0.13215
PF	0.23985	0.069956	0.48349

Table 7.5: Performance of filters.

7.3 Terrain-Aided Navigation

Performance of the DSG will be illustrated using a TAN scenarios [8], [3] and [16].

7.3.1 Cartesian Model

Let a state-space model

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}_k + \mathbf{w}_k \tag{7.9}$$

$$z_k = h(\mathbf{x}_k) + v_k \tag{7.10}$$

be considered [3]. The two-dimensional state vector $\mathbf{x}$ consists of vehicle horizontal position in north and east directions and the horizontal constant shift vector $\mathbf{u}_k = [300, 300]^T$ can be known from e.g. inertial navigation system or odometer. The noises $\mathbf{w}$ and v are described as

$$\mathcal{N}\{\mathbf{w}_k; \begin{bmatrix}0\\0\end{bmatrix} \begin{bmatrix}100 & 0\\0 & 100\end{bmatrix}\}, \tag{7.11}$$

$$\mathcal{N}\{v_k; 0, 8^2\}. \tag{7.12}$$

The measurement function h is a discrete terrain map[2] represented by a table function which assigns vertical position (i.e. altitude) to each combination of latitude and longitude it covers. The measurement z_k itself is a vehicle altitude which can be based on the barometric altimeter, radar altimeter, or their combination depending on the type of vehicle. Three PMF filters

- PMF_{TRUE} with high number of grid points $N = 7225$ providing "almost true" state estimate $p(\mathbf{x}_k|\mathbf{z}_k)$,
- PMF_{ST} with standard equidistant allocation using $N = 289$ points providing the conditional PDF $\hat{p}_{ST}(\mathbf{x}_k|\mathbf{z}_k, \boldsymbol{\xi}_k)$,
- PMF_{DSGf} using proposed DSGf using $N = 286$ points providing the conditional PDF $\hat{p}_{DSGf}(\mathbf{x}_k|\mathbf{z}_k, \boldsymbol{\xi}_k)$,

were implemented and compared in $M = 10^3$ Monte-Carlo (MC) simulations. Comparing criterions were the estimation time and integral error $\text{IE} = \frac{1}{T+1}\sum_{k=0}^{T}\int \tilde{p}(\mathbf{x}_k|\mathbf{z}^k; \boldsymbol{\xi}_k)d\mathbf{x}_k$ 3.10, where $T = 50$, both averaged over MC simulations. The results can be found in Table 7.6. This table shows significant improvement of the IE while using DSG and mild improvement

[2]The map is from Shuttle Radar Topography Mission (SRTM) an international project spearheaded by the U.S. National Geospatial-Intelligence Agency (NGA) and the U.S. National Aeronautics and Space Administration (NASA), see https://www2.jpl.nasa.gov/srtm/index.html.

of IE for DSGf over DSG. It can also be seen, as discussed in chapter 4, that better quality of PDF approximation does not have to necessarily imply more accurate mean/variance estimate. The IE for $T = 10$ can be see in Figure 7.6.

	PMF_{TRUE}	PMF_{ST}	PMF_{DSG}	PMF_{DSGf}
RMSE [m]	18.7256	18.7579	18.7271	18.7274
ASTD	22.2339	22.1840	22.2222	22.2237
IE	-	0.0223	0.0062	0.0061
N	7225	289	268	268

Table 7.6: PMFs estimation performance.

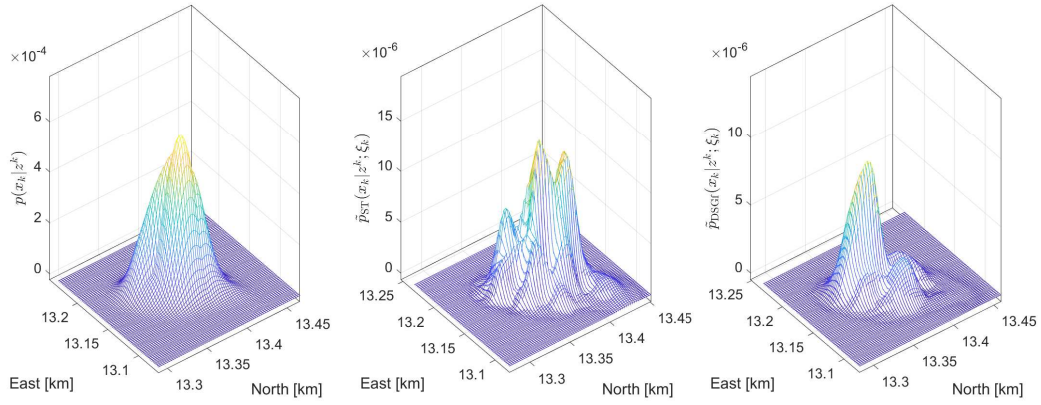

Figure 7.6: Illustration of the "true PDF" (left) and point-mas density error of standard and proposed DSGf layouts (middle, right).

7.3.2 Spherical Model

In this section the PMF was tested on a model from [16]. This model assumes constant heading course (rhumb line) with constant speed at zero altitude. The state and dynamics of the system are following

$$\mathbf{x}_k = \begin{bmatrix} \phi_k \\ \lambda_k \end{bmatrix} = \begin{bmatrix} \phi_{k-1} + (\varphi(\lambda_k) - \varphi(\lambda_{k-1}))\tan(K) \\ \lambda_{k-1} + \frac{\Delta T \|V\|}{R(\lambda_{k-1})}\cos(K) \end{bmatrix} + \mathbf{w}_k, \tag{7.13}$$

$$z_k = h(\mathbf{x}_k) + v_k, \tag{7.14}$$

where ϕ_k is latitude, λ_k is longitude, $\varphi(\lambda) = \log\left(\frac{1+\sin(\lambda)}{cos(\lambda)}\right)$, ΔT is the sampling period, state noise $\mathbf{w}_k$ is based on sensors grade, K is the constant heading relative to the north (measurement provided e.g by the compass), $\|V\|$ is a two-norm of known constant velocity vector $V = [V_{north} V_{east}]^T$ (constant heading course) and

$$R(\lambda) = \sqrt{\frac{\left(r_1^2 \cos(\lambda)\right)^2 + \left(r_2^2 \sin(\lambda)\right)^2}{\left(r_1 \cos(\lambda)\right)^2 + \left(r_2 \sin(\lambda)\right)^2}} \tag{7.15}$$

is Earth radius at latitude λ. Constants r_1 and r_2 are radius of Earth at equator and pole. The measurement function h is the same as in the preceding model but the map is in the Geographic coordinate system. The noises $\mathbf{w}$ and v are described as

$$\mathcal{N}\{\mathbf{w}_k; \begin{bmatrix}0\\0\end{bmatrix} \begin{bmatrix}1.8 \cdot 10^{-05} & 0\\ 0 & 1.8 \cdot 10^{-05}\end{bmatrix}\}, \tag{7.16}$$

$$\mathcal{N}\{v_k; 0, 3\}. \tag{7.17}$$

The results in Table 7.7 are from 100 MC, 10-time step simulations, where N is the number of points of the grid, RMSE is the square root of the MSE and ASTD is an average standard deviation of measurement step probability counted as $\sqrt{\mathsf{diag}(\mathbf{P}_k)}$ averaged over time steps and dimensions.

It can be seen that the PMF_{DSG} performs better than the PMF_{ST}. And also the PMF_{DSGf} shows better performance than standard PMF_{DSG}. The standard DDG did not perform very well in this task, but the alternative version DDG3 yielded the best result (except PMF_{TRUE}). In Figure 7.7 an example of first state varible trajectory estimate with standard derivation can be found.

	RMSE[°]	ASTD	time[s]
PMF_{TRUE}	0.017608	0.017877	1.0523
PMF_{ST}	0.021876	0.015839	0.21883
PMF_{DSG}	0.020686	0.015435	0.22041
PMF_{DSGf}	0.020372	0.015552	0.42277
PMF_{DDG}	0.022265	0.01575	0.22191
PMF_{DDG3}	0.018989	0.016562	0.23019

Table 7.7: PMFs estimation performance.

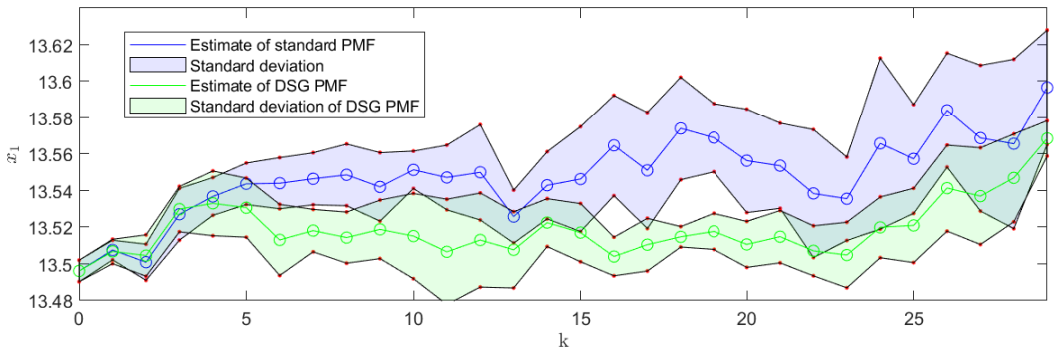

Figure 7.7: Example of a trajectory of the state x_1.

7.3.3 Comparison

The PMF DDG and DSG methods show better performance for the spherical model. It might be due to nonlinearity of the model. The decision on which model to use depends on the type of map that is available and also the sensors that can be used (e.g. compass for K, odometer for $\mathbf{u}$).

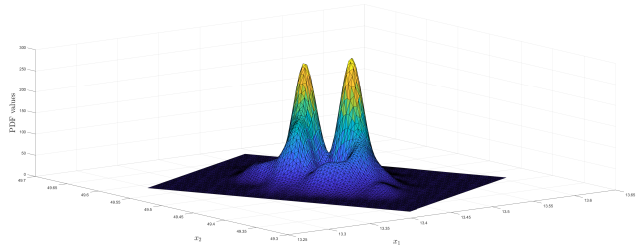

Figure 7.8: Example of predictive PDFs of spherical for $t = 20$.

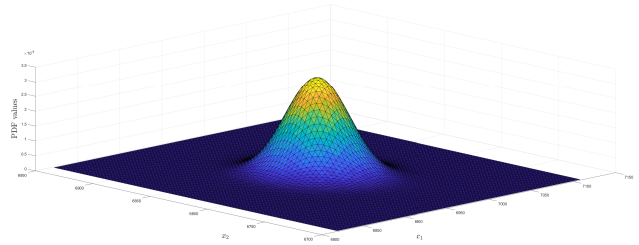

Figure 7.9: Example of predictive PDFs of Cartesian model for $t = 20$.

Chapter 8

Concluding Remarks

The first objective of this thesis was to get acquainted with the point-mass method in state estimation and navigation. After that, the aim was to design a new grid for the point-mass filter that would lead to better estimate accuracy while preserving the computational complexity. Subsequently, the PMF with the new grid design was supposed to be implemented in the non-linear estimation framework in Matlab and illustrated on a set of navigation type scenarios.

To fulfil the assignment, firstly a brief introduction and state of the art overview of the state estimation and terrain-aided navigation was presented. After that, the goal was closely introduced followed by a description of the thesis organization.

The following part offers an in-depth introduction to Bayesian estimation and subsequently covers the topic of point-mass filtering by analysing the problem of the approximation of probability density functions and point-mass solution to Bayesian recursive relations. Then a PMF algorithm is described along with multi-step prediction and smoothing.

Finally, the newly proposed density specific grid design is comprehensively described and compared to a standard grid design. Moreover, an enhanced DSG design using UKF filtration is presented. All grid designs are described with the help of theoretical algorithms.

In the following chapter, various criterion showing the quality of the PDF approximation by PMD are compared and the best parameters for the DSG are shown. There is also explored the influence of kurtosis and skewness on optimal grid parameters. The are pinpointed some of the DSG design flaws.

The findings are then used for density specific grid design enhancements, offering a new

way of setting up the outer grid boundary as well as an alternative way of creating denser grid parts.

After that, the DSG design implementation in the NEF toolbox takes place including the particular application of the hereinbefore presented standard, DSG and DSGf algorithms.

In the end, simulations were conducted in the NEF toolbox and outside, and the results illustrate the performance of DSG design PMF in comparison with the standard PMF and other filters, namely particle filter. This section also contains examples of codes showing how to use NEF.

Evaluation

It was shown that for an approximation of PDFs that have the "high value"/fast-changing part in the vicinity of their mean, it is beneficial to use the DSG design. For other PDFs, it would be probably better to use the alternative design presented in the Chapter Density Specific Grid Design Enhancements.

Future work

The DSG design may have more parameters so it can better adapt to PDFs with dominant higher moments. The parameters should be set up automatically. This could be done, for example, by a trained neural network. Also, some deeper knowledge about various approximation errors and their connections would be beneficial. Considering NEF, the DSG enhancements could be implemented.

List of Figures

Bibliography

[1] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, New Jersey, 1979.

[2] N. Bergman. Bayesian approach to terrain-aided navigation. In *Proceedings of SYSID'97*, pages 1531–1536, Kitakyushu, Fukuoka, Japan, 1997. 11th IFAC Symposium on System Identification.

[3] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. PhD thesis, Linköping University, Sweden, 1999.

[4] A. Doucet, N. De Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, 2001. (Ed. Doucet A., de Freitas N., and Gordon N.).

[5] J. Dunik. *Identifikace systemu a filtrace - skripta*. Zapadoceska univerzita v Plzni, 2018.

[6] J. Duník, O. Straka, and M. Šimandl. Stochastic integration filter. *IEEE Transactions on Automatic Control*, 58(6):1561–1566, 2013.

[7] J. Duník, O. Straka, M. Šimandl, and E. Blasch. Random-point-based filters: Analysis and comparison in target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 51(2):303–308, 2015.

[8] J. Duník, M. Soták, M. Veselý, O. Straka, and W. J. Hawkinson. Design of Rao-Blackwellised point-mass filter with application in terrain aided navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 55(1):251–272, 2019.

[9] S Edelkamp and S Schrödl. Chapter 17 - vehicle navigation. In *Heuristic Search*, pages 737 – 757. Morgan Kaufmann, San Francisco, 2012. ISBN 978-0-12-372512-7. doi: https://doi.org/10.1016/B978-0-12-372512-7.00017-1. URL `http://www.sciencedirect.com/science/article/pii/B9780123725127000171`.

[10] M. Flídr, O. Straka, and M. Šimandl. Nonlinear estimation software framework in optimal and adaptive control problems. In *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, August 2014.

[11] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *IEEE Proceedings*, 92(3):401–421, 2004.

[12] O. Kost, O. Straka, and J. Duník. Identification of state and measurement noise covariance matrices using Nonlinear Estimation Framework. In *Proceedings of the 12th European Workshop on Advanced Control and Diagnosis*, Pilsen, Czech Republic, November 2015.

[13] J. Královec and M. Šimandl. Filtering, prediction and smoothing with point-mass approach. In *Preprints of the 16th IFAC Symposium on Automatic Control in Aerospace*, Saint Petersburg, Russia, 2004.

[14] J. Královec and M. Šimandl. Filtering, prediction and smoothing with point-mass approach. In *Automatic Control in Aerospace 2004*, pages 375–380, Oxford, 2005. Elsevier.

[15] J. Matoušek, J. Duník, and O. Straka. Point-mass fitler: Density specific grid design and implementation,. In *15th European Workshop on Advanced Control and Diagnosis*, Bologna, Italy, 2019.

[16] C. Musso, A. Bresson, Y. Bidel, N. Zahzam, K. Dahia, J. Allard, and B. Sacleux. Absolute gravimeter for terrain-aided navigation. In *2017 20th International Conference on Information Fusion (Fusion)*, pages 1–7, July 2017. doi: 10.23919/ICIF.2017.8009805.

[17] A. Papoulis and S. U. Pillai. *Probability, Random Variables and Stochastic Processes*. Mc Graw Hill, fourth edition edition, 2002.

[18] D. Peng, T. Zhou, C. Xu, Zhang W., and J. Shen. Marginalized point mass filter with estimating tidal depth bias for underwater terrain-aided navigation. *Journal of Sensors*, 2019, 2019.

[19] S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[20] N. Sirola. *Nonlinear filtering with piecewise probability densities*. PhD thesis, Tampere Univ. of Technology, Finland, 2007.

[21] H. W. Sorenson. On the development of practical nonlinear filters. *Information Sciences*, 7:230–270, 1974.

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

[23] M. Šimandl and J. Duník. Derivative-free estimation methods: New results and performance analysis. *Automatica*, 45(7):1749–1757, 2009.

[24] M. Šimandl, J. Královec, and T. Söderström. Anticipative grid design in point-mass approach to nonlinear state estimation. *IEEE Transactions on Automatic Control*, 47 (4), 2002.

[25] M. Šimandl, J. Královec, and T. Söderström. Advanced point-mass method for nonlinear state estimation. *Automatica*, 42(7):1133–1145, 2006.

[26] V. Šmídl and M. Gašperin. Rao-Blackwellized point mass filter for reliable state estimation. In *Proceedings of the 16th International Conference on Information Fusion*, Istanbul, Turkey, 2013.

[27] Y. M. Yoo and C. G. Park. Improvement of terrain referenced navigation using a point mass filter with grid adaptation. *International Journal of Control, Automation, and Systems*, 13(5):1173–1181, 2015.

Appendix

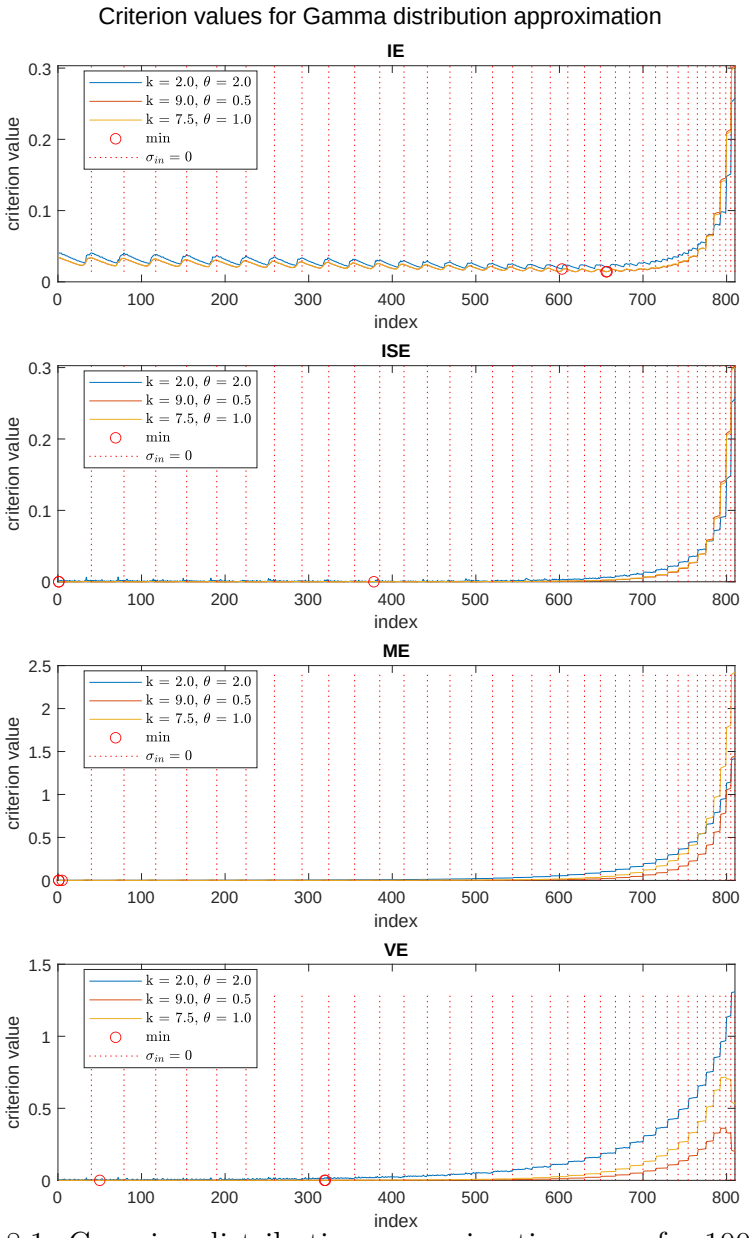

Figure 8.1: Gaussian distribution approximation error for 100 points.

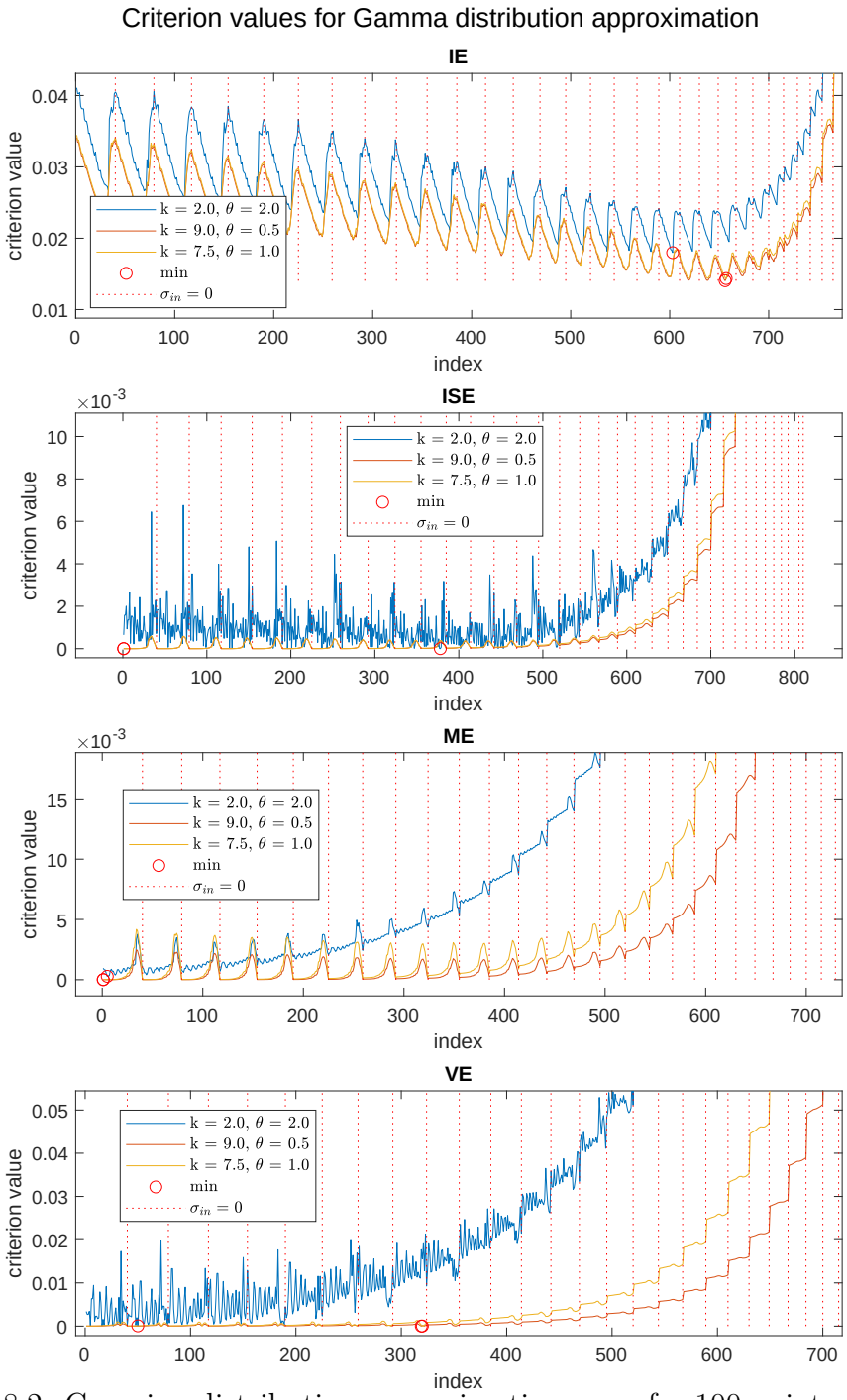

Figure 8.2: Gaussian distribution approximation error for 100 points - zoomed.

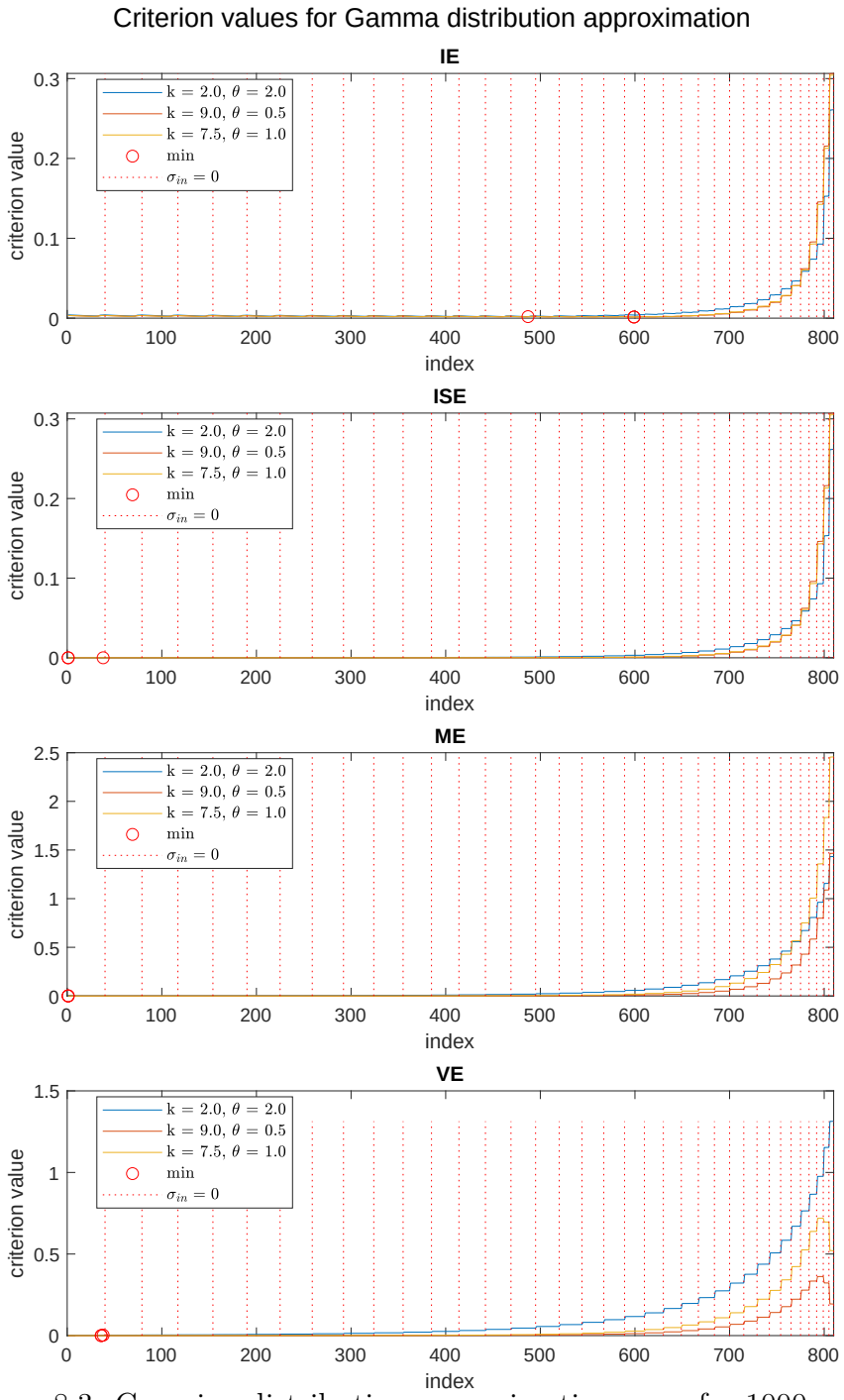

Figure 8.3: Gaussian distribution approximation error for 1000 points.

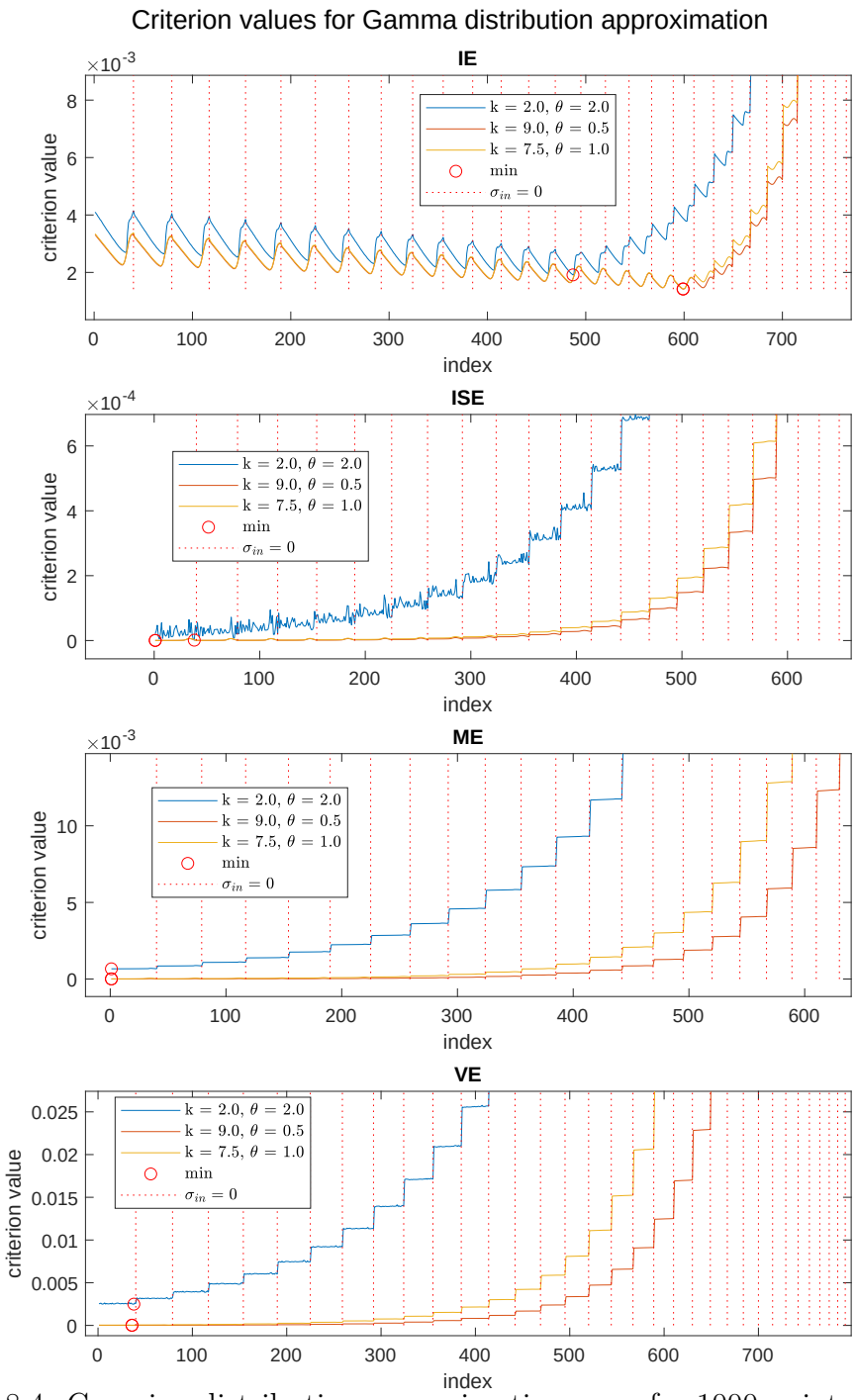

Figure 8.4: Gaussian distribution approximation error for 1000 points - zoomed.