Unscented Kalman Filter using Optimal Quantization

Vivien Pravong, Jean-Philippe Condomines, Gustav Öman Lundin, Stéphane Puechmorel

Unscented Kalman Filter using Optimal Quantization

Vivien Pravong[1,3], Jean-Philippe Condomines[2,3], Gustav Öman Lundin[1,3] and Stéphane Puechmorel[2,3]
[1]DTIS, ONERA, Université de Toulouse, 31055 Toulouse, France
firstname.lastname@onera.fr
[2]Ecole Nationale de l'Aviation Civile (ENAC), Université de Toulouse, 31055 Toulouse, France
firstname.lastname@enac.fr
[3]Fédération ONERA ISAE-SUPAERO ENAC Université de Toulouse, France

***Abstract*—This paper presents a novel approach to deal with nonlinear filtering by augmenting an Unscented Kalman Filter (UKF) with an Optimal quantization algorithm, named OQ-UKF. The Unscented Kalman Filter uses a sigma-point based method to approximate the distribution of an unknown random variable onto which is applied a nonlinear transformation, providing a cloud of evolving points. However, the generation of these so-called sigma-points is done by a deterministic algorithm which needs tuning in order to accurately capture the distribution of the estimate. This tuning is often problem-dependent due to nonlinearities and sometimes not optimal. We propose to fuse an UKF with Optimal quantization whose objective is to find the best approximation of the density of a random variable. The designed OQ-UKF is described in this paper, and its performance is evaluated for some relevant practical problems, such as pose estimation of a two-dimensional mobile robot.**

***Index Terms*—Bayesian Estimation, Unscented Transform, Optimal Quantization, Unscented Kalman Filter**

I. Introduction

Stochastic observers such as the Kalman filters are commonly used to solve state estimation problems. Notably, the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) [1] are the two main versions when dealing with nonlinear problems. It has been shown that the UKF performs better than the EKF in general. This is because the UKF does not rely on any computation of the Jacobians of the nonlinear function but uses an Unscented Transform (UT) to propagate the state uncertainty. The UT can be interpreted as a numerical computation of the derivatives [2]. The idea is to spread a finite number of points around the estimate, called the sigma-points (σ-points), then propagate them through the nonlinear function to capture the mean and the covariance accurately to the third order (of the Taylor series expansion). On the contrary, the EKF only approximates the Taylor series up to the first order in general. Thus, the UKF achieves a better level of accuracy at a similar complexity as that of the EKF [3].

Regarding the UT, it is necessary to tune some hyperparameters in order to accurately capture the distribution of the estimate. These parameters, often noted as α, β and κ (also known as scaling parameters [4]), allow us to tune the spreading of the σ-points around the estimate, as well as generating the weights used to compute the mean and the covariance. Many guidelines exist to adequatly tune these parameters but none of them provides the best tuning for all problems. In this paper, we suggest an alternative approach relying on Optimal quantization.

A. Links with previous literature

To date, only a few studies have attempted to address the problem of systematically finding optimal parameters [6]–[10]. Some of these works employed data-driven methods to learn these parameters, such as discriminative training methods [8] or genetic algorithms [9]. However, these approaches require a substantial amount of data for training and remain specific to particular applications. Other studies tackled this issue by formulating and solving an optimization problem. This is the case in [7] and [10], where the objective is to maximize the log-likelihood of the measurements given a set of parameters α, β and κ. In [10], a guideline was proposed to guide the resolution of the optimization problem, but only in the scalar case, while the multivariate case was briefly addressed. No further investigation in line with this study has been conducted until now. As demonstrated in [11], tuning is problem-dependent since various types of nonlinearities can influence the selection of these parameters. Apart from these methods, their values are often left unchanged from the early publications [1], [3], [5]. Essentially, three sets of parameters can be distinguished: the first set proposed in [1] relies solely on κ. The coefficients α and β were introduced later, and their suggested values are given in [3]. The third set is derived from the cubature rule provided in [5]. Although these sets may differ by their respective values, the commonality among all of them is that they are based on polynomial approximation.

Regarding Optimal quantization, the term "quantization" takes root from the field of signal processing, where the goal is to discretize a continuous signal to address transmission issues. In the context of probabilities, quantization involves a set of methods to approximate a continuous n-dimensional probability density with a discrete probability containing N supporting points [13]. Optimal quantization, in particular, delves into refining this process to achieve the most accurate representation of the original continuous distribution with the least possible error. The primary objective is to determine

the best placement of discrete points in the probability space based on an optimal approximation of a probability measure. This method is particularly useful in nonlinear filtering and Bayesian fusion, where the goal is to compute the conditional probability of a nonlinear stochastic process given past informations. In [14], an Optimal quantization approach has been developed to numerically solve nonlinear filtering problems associated with discrete-time observations. Another significant contribution by the same authors explores the application of Optimal quantization specifically when the underlying probability density is Gaussian [15]. While there exists some applications in the field of finance [16], [17], to the best of our knowledge, very few applications of Optimal quantization (with the optimality criterion as defined in [14], [15]) for Kalman filtering in other field (e.g. robotics) exist. Some works can be interpreted as adopting an Optimal quantization approach [18] but based on another optimality criterion which is the minimization of the Cramer-Von Mises distance [19].

The contribution of this paper is the development of a novel method, that we refer to as OQ-UKF, which is based on both Unscented Transform and Optimal quantization (with the optimality criterion of [14]). In this paper, we are not interested in the scaling parameters tuning but we provide an additional support to the UT.

B. Paper's organisation

The paper is organized as follows: in section II, mathematical tools are provided. In section III, we recall the problem of Bayesian estimation, followed by the Unscented Transform procedure used to address the problem. A description of the new approach using the Optimal quantization is then given. In section IV, we describe the OQ-UKF resulting from our approach and present some simulation results in section V in the context of pose estimation. We draw some conclusions and remarks in section VI.

II. Mathematical preliminaries

In this section, we recall basic definitions and properties of the quadrature formula and Optimal quantization.

A. Quadrature formula

Consider a random vector $\mathbf{x} \in \mathbb{R}^n$ equipped with a probability measure $d\mu(\mathbf{x}) = \mathcal{P}(\mathbf{x})d\mathbf{x}$ where $\mathcal{P}(\mathbf{x})$ is the probability density of $\mathbf{x}$. A random vector is transformed through any function $g(.)$ and the moment integral is given by

$$\mathbb{E}(g(\mathbf{x})) = \int_{\mathbb{R}^n} g(\mathbf{x})d\mu(\mathbf{x}) \tag{1}$$

To numerically compute (1), one can rely on a finite set of points $\{\mathbf{x}_i\}_{i=1}^N$ (also called nodes) and weights $\{w_i\}_{i=1}^N$ such that the measure $\mu(\mathbf{x})$ is replaced by a finite sum of Dirac measures $\sum_{i=1}^N w_i\delta_{\mathbf{x}_i}(\mathbf{x})$. Locally about the point $\mathbf{x}$, the Dirac measure can be defined as

$$\forall A \in \mathcal{P}(\Omega), \delta_{\mathbf{x}}(A) = \begin{cases} 0 & \mathbf{x} \notin A \\ 1 & \mathbf{x} \in A \end{cases}$$

where Ω is the measure space on which the integral is defined.

This replacement gives rise to an approximation of (1) as

$$\int_{\mathbb{R}^n} g(\mathbf{x})d\mu(\mathbf{x}) \approx \sum_{i=1}^N w_i g(\mathbf{x}_i) \tag{2}$$

with the condition $\sum_{i=1}^N w_i = 1$. In the quadrature formula (2), the set of weights and nodes must represent the moments of the given distribution as exact as possible.

B. Optimal quantization

An N-quantizer is defined as a mapping $\mathcal{Q}\colon \mathbb{R}^n \rightarrow \{\mathbf{x}_1, ..., \mathbf{x}_N\}$. Let $||.||_p$ denotes the l^p-norm ($p \geq 1$). The N-quantizer $\mathcal{Q}$ is said to be p-optimal if it minimizes the expected value of the quantization error (also called *distortion*):

$$\mathbb{E}(||\mathbf{x} - \mathcal{Q}(\mathbf{x})||_p) = \int_{\mathbb{R}^n} ||\mathbf{x} - \mathcal{Q}(\mathbf{x})||_p d\mu(\mathbf{x}) \tag{3}$$

An N-quantizer that would be p-optimal for (3) is the Voronoï quantizer $\mathcal{Q}_{vor}\colon \mathbb{R}^n \rightarrow \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ defined as:

$$\mathcal{Q}_{vor}(\mathbf{x}) = \sum_{i=1}^N \mathbf{x}_i \delta_{C(\mathbf{x}_i)}(\mathbf{x}) \tag{4}$$

where $C(\mathbf{x}_i)$ is the i^{th} Voronoï cell such that:

$$C(\mathbf{x}_i) \subset \{\mathbf{x} \in \mathbb{R}^n : ||\mathbf{x} - \mathbf{x}_i||_p \leq ||\mathbf{x} - \mathbf{x}_j||_p,\ j = \overline{1, N}\}$$

This quantizer aims to place the set of points $\{\mathbf{x}_i\}_{i=1}^N$ such that it creates a Voronoï tesselation (see [13] for more details).

III. Bayesian estimation and moment integrals

Consider a state vector $\mathbf{x}_t \in \mathbb{R}^n$, with prior probability $\mathcal{P}(\mathbf{x}_t)$ at a given instant t. Assume we get some information about $\mathbf{x}_t$ through a measurement $\mathbf{y}_t \in \mathbb{R}^m$ (with $m \leq n$). The goal of Bayesian estimation is to compute an estimate of the posterior probability $\mathcal{P}(\mathbf{x}_t|\mathbf{y}_t)$ at each time step. In subsection III-A, we recall the integrals needed to solve the Bayesian estimation problem as well as some weaknesses regarding the approximation of these integrals by the UT. We then describe a proposition of solution in III-B and provide preliminary results in III-C.

A. Moments approximation by Unscented Transform

Consider a generic observation vector of the form

$$\mathbf{y} = g(\mathbf{x}) + \mathbf{v} \tag{5}$$

where $g(.)$ is a known nonlinear function and $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \in \mathbb{R}^m$ is a white zero-mean Gaussian noise. The problem of Bayesian estimation is as follows:

1) Given a known initial conditional probability

$$\mathcal{P}(\mathbf{x}_0|\mathbf{y}_0) = \mathcal{P}(\mathbf{x}_0) \tag{6}$$

2) Assume an observation $\mathbf{y}_t$ is available

3) Compute recursively, the posterior probability as

$$\mathcal{P}(\mathbf{x}_t|\mathbf{y}_t) = \frac{\mathcal{P}(\mathbf{y}_t|\mathbf{x}_t)\mathcal{P}(\mathbf{x}_t|\mathbf{y}_{t-1})}{\mathcal{P}(\mathbf{y}_t|\mathbf{y}_{t-1})} \tag{7}$$

with

$$\mathcal{P}(\mathbf{x}_t|\mathbf{y}_{t-1}) = \int_{\mathbb{R}^n} \mathcal{P}(\mathbf{x}_t|\mathbf{x}_{t-1})\mathcal{P}(\mathbf{x}_{t-1}|\mathbf{y}_{t-1})d\mathbf{x}_{t-1} \quad (8)$$

and

$$\mathcal{P}(\mathbf{y}_t|\mathbf{y}_{t-1}) = \int_{\mathbb{R}^n} \mathcal{P}(\mathbf{y}_t|\mathbf{x}_t)\mathcal{P}(\mathbf{x}_t|\mathbf{y}_{t-1})d\mathbf{x}_t \quad (9)$$

Assume that we have $\mathbf{x} \sim \mathcal{N}(\overline{\mathbf{x}}, \mathbf{P})$. Then, the integrals in (8-9) can be reduced to integrals of the form of (1) with $d\mu(\mathbf{x}) = \mathcal{N}(\overline{\mathbf{x}}, \mathbf{P})d\mathbf{x}$ and can be computed using (2).

To attack this problem, there exists two approaches:

(i) the weights and points locations are chosen so that (2) is exact for all g that can be approximated by a set of polynomials of degree up to a given integer.

(ii) the weights and points locations are chosen based on a closeness criterion between the true and the discrete measure $\mu(\mathbf{x})$.

The Unscented Transform pertains to the first category, as it relies on the Gauss quadrature methods [20] [21] with N points, which aims to make (2) exact for all polynomials of degree up to $2N-1$. The UT approximates the posterior $p(\mathbf{x}_t|\mathbf{y}_t)$ as follows: one generate a finite number of samples $\{\boldsymbol{\chi}_i\}_{i=0}^{2n}$ and then pass each of these so-called σ-points through (5). Then, one can compute successively the mean $\overline{\mathbf{y}} = \mathbb{E}[\mathbf{y}]$ and the covariance $\mathbf{P_y} = \mathbb{E}[(\mathbf{y}-\overline{\mathbf{y}})(\mathbf{y}-\overline{\mathbf{y}})^T]$ using the quadrature formulas:

$$\begin{aligned}
\overline{\mathbf{y}} &= \int g(\mathbf{x})\mathcal{N}(\overline{\mathbf{x}}, \mathbf{P})d\mathbf{x} \approx \sum_{i=0}^{2n} W_i^{(m)} g(\boldsymbol{\chi}_i) \\
\mathbf{P_y} &= \int (g(\mathbf{x})-\overline{\mathbf{y}})(g(\mathbf{x})-\overline{\mathbf{y}})^T \mathcal{N}(\overline{\mathbf{x}}, \mathbf{P})d\mathbf{x} + \mathbf{R} \quad (10) \\
&\approx \sum_{i=0}^{2n} W_i^{(c)} (g(\boldsymbol{\chi}_i)-\overline{\mathbf{y}})(g(\boldsymbol{\chi}_i)-\overline{\mathbf{y}})^T + \mathbf{R}
\end{aligned}$$

In (10), the coefficients $\{W_i^{(m)}\}_{i=0}^{2n}$ and $\{W_i^{(c)}\}_{i=0}^{2n}$ constitute the weights. The σ-points are deterministically generated using the mean $\overline{\mathbf{x}}$, the covariance $\mathbf{P}$ and the scaling parameters α, β, κ. The general framework is

$$\begin{aligned}
\boldsymbol{\chi}_0 &= \overline{\mathbf{x}} \\
\boldsymbol{\chi}_i &= \overline{\mathbf{x}} - \sqrt{n+\lambda}\mathbf{S}_i \;\; \forall i = \overline{1,n} \quad (11) \\
\boldsymbol{\chi}_i &= \overline{\mathbf{x}} + \sqrt{n+\lambda}\mathbf{S}_i \;\; \forall i = \overline{1,n}
\end{aligned}$$

where $\lambda = \alpha^2(n+\kappa) - n$ and $\mathbf{S}_i$ denotes the i^{th} column of the matrix $\mathbf{S}$ which is a decomposition of the covariance matrix such that $\mathbf{S}^T\mathbf{S} = \mathbf{P}$. The corresponding weights are

$$\begin{aligned}
W_0^{(m)} &= \frac{\lambda}{n+\lambda}\,,\; W_0^{(c)} = \frac{\lambda}{n+\lambda} + (1-\alpha^2+\beta) \\
W_i^{(m)} &= W_i^{(c)} = \frac{1}{2(n+\lambda)}\,,\; \forall i = \overline{1,2n}
\end{aligned} \quad (12)$$

Finding the optimal set of points and weights via the scaling parameters is not straightforward and heavily depends on the problem at stake. As a matter of fact, the scaling-parameters are often hand-tuned but generally, one can sort out three main sets which we refer to as UT1 [1], UT2 [3] and CT (Cubature Transform) [5]. (See Table III in Appendix A for the difference between each set).

In this paper, we motivate the need to improve one of these standard UTs by giving a first insight on their performances. To simplify, let's consider a Gaussian random vector $\mathbf{x} \in \mathbb{R}^2$ with known statistics $\overline{\mathbf{x}}$ and $\mathbf{P}$. This random vector is transformed to a scalar random variable y by

$$y = \cos^2(x_1) + \sin^2(x_2) \quad (13)$$

for which we seek to compute the moments $\overline{y}$ and P_y. We can numerically show that the UT gives weak results. In Table I, we have listed the absolute difference between the true and approximated moments obtained with the different transformations. The true moments are obtained thanks to a Monte-Carlo Transformation (MCT).

TABLE I: Absolute difference between the true (indexed by MC) and approximated moments. The number of samples for the Monte-Carlo Transformation is 100 000. The prior moments are $\overline{\mathbf{x}} = [0, \frac{\pi}{2}]^T$ and $\mathbf{P} = 2\mathbf{I}_{2\times 2}$.

	$\lvert\overline{y}_{MC} - \overline{y}\rvert$	$\lvert(P_y)_{MC} - P_y\rvert$
UT1 [1]	0.7102	0.2129
UT2 [3]	3.0183	31.7502
CT [5]	0.1549	0.2498

It seems that the moments given by the different UTs could be better, as demonstrated by the errors in Table I. This was expected since the UT approximates the moments up to a certain degree. The core of what we address in this paper is how to get the best set of points and weights in order to make the approximations in (10) as exact as possible. We propose to tackle this problem by coupling the UT with Optimal quantization.

B. Approach using Optimal quantization

The generation of the σ-points in the Unscented Transform is typically a $(2n+1)$-quantizer of the probability density of $\mathbf{x}$. However it may not be optimal in the sense of (3). As stated in the mathematical preliminaries, the p-optimal quantizer is the Voronoï quantizer. Our goal is to readjust the σ-points to yield a new set that would be closer to a Voronoï tesselation.

Starting from a suboptimal set of points $\{\mathbf{x}_i\}_{i=0}^{2n}$, one can converge to an optimal one by minimizing the distortion

$$\mathbb{E}(||\mathbf{x} - \mathcal{Q}(\mathbf{x})||_p) = \sum_{i=0}^{2n} \int_{\mathbb{R}^n} ||\mathbf{x} - \mathbf{x}_i||_p \mathcal{N}(\overline{\mathbf{x}}, \mathbf{P})d\mathbf{x} \quad (14)$$

The next proposition is proved in [15].

Proposition 1. *Equation (14) is continuously differentiable and admits a minimizer if the points are distinct pairwise* $\mathbf{x}_i \neq \mathbf{x}_j$ *for* $i \neq j$.

In the same reference, an algorithm based on a stochastic gradient descent is proposed to solve the minimization of (14). The procedure is described in Algorithm 1.

Algorithm 1 Competitive Learning Vector Quantization (CLVQ)

Inputs: $\{\mathbf{x}_i\}_{i=0}^{2n}$, $\overline{\mathbf{x}} \in \mathbb{R}^n$, $\mathbf{P} \in \mathbb{R}^{n\times n}$, $k_{max} \in \mathbb{N}$, $\{\gamma_k\}_{k=1}^{k_{max}}$

for $k = 1$ to k_{max} **do**
  Draw $\mathbf{x}_{rand}$ from probability distribution $\mathcal{N}(\overline{\mathbf{x}}, \mathbf{P})$
  Find the winning index $j = \underset{i}{\text{argmin}}(||\mathbf{x}_{rand} - \mathbf{x}_i||_p)$
  $\mathbf{x}_i \leftarrow \mathbf{x}_i \ \forall i \neq j$
  $\mathbf{x}_j \leftarrow \mathbf{x}_j - \gamma_k(\mathbf{x}_j - \mathbf{x}_{rand})$
end for

Outputs: $\{\mathbf{x}_i\}_{i=0}^{2n}$

The steps γ_k are elements of a sequence $\{\gamma_k\}, k \in \mathbb{N}$ satisfying:

$$\begin{aligned} &\gamma_k \in]0,1[\\ &\sum_k \gamma_k = +\infty, \sum_k \gamma_k^2 < +\infty \end{aligned} \tag{15}$$

Since the UT already provides a set of distinct points $\boldsymbol{\chi}_i$, our idea is to apply the CLVQ algorithm on this set to yield a new placement which would be closer to the optimal placement in the sense of (14). In contrast to the usual approach in the literature, we propose to start from one of the suggested values proposed in Table III to generate a prior set of σ-points and weights. We then readjust the σ-points using the Algorithm 1. By doing so, we realize a compromise between Gauss quadrature methods, which rely on approach (i) and the approximation of the probability measure relying on approach (ii).

Furthermore, as explained in [15], one can compute new weights (also called *μ-masses* of the Voronoï cells) as a by-product during the CLVQ procedure. To do that, one can simply initialize a counter $q_i = 0$ for each point i and then increment the j^{th} counter each time the j^{th} point is selected in the process: $q_j \leftarrow q_j + 1$ where j is the winning index. At the end of the loop, the weights $\{w_i\}_{i=0}^{2n}$ are obtained by computing $w_i = q_i/k_{max}\ \forall i$. However in this paper, we decided to consider the points placement only and keep the original set of weights. The reasons are given in section VI.

C. First results

Let's consider again the nonlinear function (13). We propose to apply the Algorithm 1 on the σ-points generated by the UT1 and the CT sets only[1]. The Algorithm 1 takes place before the propagation into (13). The total number of random draw was $k_{max} = 10000$ for both transformations and a sequence $\{\gamma_k\}_{k=1}^{k_{max}}$ satisfying (15) was chosen as $\gamma_k = (\frac{1}{10}\sqrt{\mathbf{P}})/k$ for the UT1 and $\gamma_k = (\frac{1}{4}\sqrt{\mathbf{P}})/k$ for the CT.

Since high uncertainty situations are precisely where one would want moments approximation to be particularly accurate, we evaluated the performance for different $\mathbf{P}$. In Fig. 1, we have displayed the absolute error (averaged on a dozen simulations) between the true and approximated moments of y given by the different UTs in function of $||\mathbf{P}||_F$ where $||.||_F$ denotes the Frobenius norm. As we can see for small $\mathbf{P}$, our method does not provide any particular improvement (it is sometimes slightly deteriorated) but the error begins to decrease as the uncertainty on $\mathbf{x}$ grows. At a certain point, our approach outperforms the standard method for both UTs although the evolution follows the same tendency. Additional results are provided in Appendix B regarding the evolution of the distortion.

Overall, the σ-points placement provided by the Unscented Transform and corrected by the CLVQ is either identical or better than the initial placement when the uncertainty is high. This example illustrates and motivates our proposition to fuse an UKF with an Optimal quantization algorithm.

[1]We chose to put aside the UT2 set since it provided less good results than the others.

IV. Application to Unscented Kalman Filtering

In this section, we apply our approach on the Unscented Kalman Filter and refer to it as the OQ-UKF. Consider the following discrete-time nonlinear stochastic system

$$\begin{aligned} \mathbf{x}_t &= f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) + \mathbf{w}_{t-1} \\ \mathbf{y}_t &= h(\mathbf{x}_t) + \mathbf{v}_t \end{aligned} \tag{16}$$

where $\mathbf{x}_t \in \mathbb{R}^n$ corresponds to the state of the system, $\mathbf{y}_t \in \mathbb{R}^m$ is a noisy measurement and $\mathbf{u}_t \in \mathbb{R}^p$ is a known deterministic input. $f\colon \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ and $h\colon \mathbb{R}^n \to \mathbb{R}^m$ are known nonlinear functions denoting the dynamical and the observation model respectively. For notation convenience, we will now write $f(\mathbf{x}_t, \mathbf{u}_t) = f(\mathbf{x}_t)$. $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$ is the model noise and $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$ is the measurement noise.

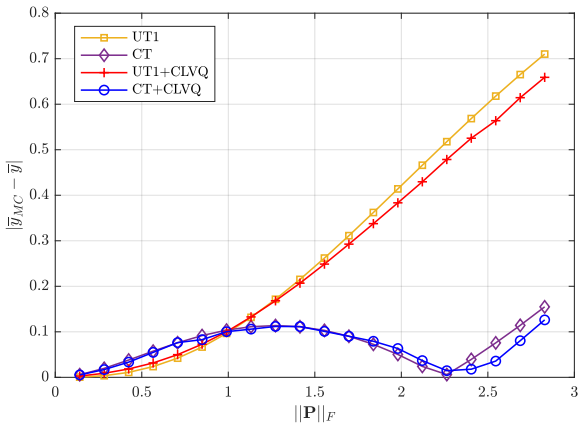
Kalman filters are part of the family of Gaussian filters [22]. The general form of a Gaussian filter comes from the Bayesian estimator as it solves the estimation problem by a recursive algorithm consisting on a prediction and update step, under the assumption that $\mathbf{x}$ is Gaussian. Assuming that $\overline{\mathbf{x}}_{t-1|t-1}$ and $\mathbf{P}_{t-1|t-1}$ are the mean and covariance at time $t-1$ given measurements at $t-1$, the filtering process is given as follows:

1) Prediction:

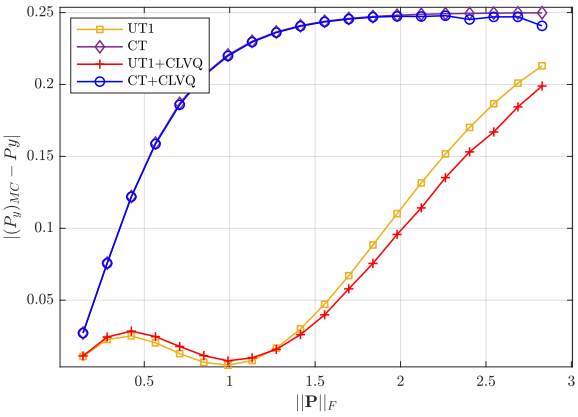
$$\begin{aligned} \overline{\mathbf{x}}_{t|t-1} &= \int f(\mathbf{x}_{t-1})\mathcal{N}(\overline{\mathbf{x}}_{t-1|t-1}, \mathbf{P}_{t-1|t-1})d\mathbf{x}_{t-1} \\ \mathbf{P}_{t|t-1} &= \int (f(\mathbf{x}_{t-1}) - \overline{\mathbf{x}}_{t|t-1})(f(\mathbf{x}_{t-1}) - \overline{\mathbf{x}}_{t|t-1})^T ... \\ &\quad ... \times \mathcal{N}(\overline{\mathbf{x}}_{t-1|t-1}, \mathbf{P}_{t-1|t-1})d\mathbf{x}_{t-1} + \mathbf{Q}_{t-1} \end{aligned} \tag{17}$$

2) Update:

$$\begin{aligned} \overline{\mathbf{y}}_t &= \int h(\mathbf{x}_t)\mathcal{N}(\overline{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1})d\mathbf{x}_t \\ \mathbf{P}_{\mathbf{y}_t} &= \int (h(\mathbf{x}_t) - \overline{\mathbf{y}}_t)(h(\mathbf{x}_t) - \overline{\mathbf{y}}_t)^T ... \\ &\quad ... \times \mathcal{N}(\overline{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1})d\mathbf{x}_t + \mathbf{R}_t \\ \mathbf{P}_{\mathbf{x}\mathbf{y}_t} &= \int (\mathbf{x}_t - \overline{\mathbf{x}}_{t|t-1})(\mathbf{y}_t - \overline{\mathbf{y}}_t)^T ... \\ &\quad ... \times \mathcal{N}(\overline{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1})d\mathbf{x}_t \\ \mathbf{K}_t &= \mathbf{P}_{\mathbf{x}\mathbf{y}_t}\mathbf{P}_{\mathbf{y}_t}^{-1} \\ \overline{\mathbf{x}}_{t|t} &= \overline{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_m - \overline{\mathbf{y}}_t) \\ \mathbf{P}_{t|t} &= \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{P}_{\mathbf{y}_t}\mathbf{K}_t^T \end{aligned} \tag{18}$$



(a) Absolute error between the true mean $\overline{y}_{MC}$ and the approximated mean $\overline{y}$ as a function of $||\mathbf{P}||_F$



(b) Absolute error between the true variance $(P_y)_{MC}$ and the approximated variance P_y as a function of $||\mathbf{P}||_F$

Fig. 1: Mean and variance errors of y for various prior covariances $\mathbf{P}$ (with $\overline{\mathbf{x}} = [0, \pi/2]^T$ fixed)

In the equations (18), $\overline{\mathbf{y}}_t$ is the mean of the predicted measurement and $\mathbf{y}_m$ is the observation acquired at time t. $\mathbf{P}_{\mathbf{y}_t}$, $\mathbf{P}_{\mathbf{x}\mathbf{y}_t}$ and $\mathbf{K}_t$ are respectively the covariance of the predicted measurement, the cross-covariance and the Kalman gain.

The UKF uses the UT to compute the moment integrals in (17) and (18). We propose to compute these integrals thanks to our approach to yield the UKF described in Algorithm 2. This algorithm consists in applying the CLVQ each time one generate σ-points. This procedure is added in steps 3 and 7, where the σ-points are readjusted using the available statistical properties of $\mathbf{x}_t$ as inputs for Algorithm 1.

V. Simulation results for pose estimation

In this section, we evaluate our method on a pose estimation problem for a mobile robot using simulated data on Matlab.

A. System model

The considered system is a 2D rigid body. Let the vector state $\mathbf{X} = [\theta, x_1, x_2]^T$ where θ denotes the heading and $\mathbf{x} = [x_1, x_2]^T$ the position w.r.t a fixed reference frame. The discrete kinematical model describing the motion of the system from time $t-1$ to t is given as follows:

$$\begin{aligned} \theta_t &= \theta_{t-1} + (u^\theta_{t-1} + w^\theta_{t-1})\Delta t \\ \mathbf{x}_t &= \mathbf{x}_{t-1} + \begin{pmatrix} \cos(\theta_{t-1}) & -\sin(\theta_{t-1}) \\ \sin(\theta_{t-1}) & \cos(\theta_{t-1}) \end{pmatrix} \begin{bmatrix} u^1_{t-1} + w^1_{t-1} \\ u^2_{t-1} + w^2_{t-1} \end{bmatrix} \Delta t \end{aligned} \tag{19}$$

Where $u^\theta_t \in \mathbb{R}$ denotes the angular velocity, $u^1_t \in \mathbb{R}$ and $u^2_t \in \mathbb{R}$ respectively represent the longitudinal and transversal linear velocities (both expressed in the body frame). The quantities w^θ_t, w^1_t and w^2_t are the model uncertainties on the angular velocity, longitudinal shift and transversal shift respectively. Δt is a fixed time step. These variables are gathered in the vectors:

$\mathbf{u}_t = [u^\theta_t,\ u^1_t,\ u^2_t]^T$

$\mathbf{w}_t = [w^\theta_t,\ w^1_t,\ w^2_t]^T \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q} = \text{diag}[\sigma^2_\theta, \sigma^2_1, \sigma^2_2]\right)$

Algorithm 2 OQ-UKF

Inputs: $\hat{\mathbf{x}}_{0|0}$, $\mathbf{P}_{0|0}$, $\mathbf{Q}$, $\mathbf{R}$, α, β, κ, k_{max}, $\{\gamma_k\}_{k=1}^{k_{max}}$

1: **for** $t = 1$ to t_{end} **do**

2: Generate σ-points $\{\boldsymbol{\chi}_i\}_{i=0}^{2n}$ with $\overline{\mathbf{x}} = \hat{\mathbf{x}}_{t-1|t-1}$, $\mathbf{S} = \mathbf{S}_{t-1|t-1}$ using (11) and weights $\{W_i^{(m)}, W_i^{(c)}\}_{i=0}^{2n}$ using (12)

3: Readjust σ-points using Algorithm 1 by taking $\overline{\mathbf{x}} = \hat{\mathbf{x}}_{t-1|t-1}$ and $\mathbf{P} = \mathbf{P}_{t-1|t-1}$

4: Compute the predicted mean and covariance:
$\hat{\mathbf{x}}_{t|t-1} = \sum_{i=0}^{2n} W_i^{(m)} f(\boldsymbol{\chi}_i)$
$\mathbf{P}_{t|t-1} = \sum_{i=0}^{2n} W_i^{(c)} (f(\boldsymbol{\chi}_i) - \hat{\mathbf{x}}_{t|t-1})(f(\boldsymbol{\chi}_i) - \hat{\mathbf{x}}_{t|t-1})^T + \mathbf{Q}$

5: **if** measurement $\mathbf{y}_m$ acquired **then**

6: Generate σ-points $\{\boldsymbol{\chi}_i\}_{i=0}^{2n}$ with $\overline{\mathbf{x}} = \hat{\mathbf{x}}_{t|t-1}$, $\mathbf{S} = \mathbf{S}_{t|t-1}$ using (11) and weights $\{W_i^{(m)}, W_i^{(c)}\}_{i=0}^{2n}$ using (12)

7: Readjust σ-points using Algorithm 1 by taking $\overline{\mathbf{x}} = \hat{\mathbf{x}}_{t|t-1}, \mathbf{P} = \mathbf{P}_{t|t-1}$

8: Apply the correction step:
$\hat{\mathbf{y}}_t = \sum_{i=0}^{2n} W_i^{(m)} h(\boldsymbol{\chi}_i)$
$\mathbf{P}_{\mathbf{y}_t} = \sum_{i=0}^{2n} W_i^{(c)} (h(\boldsymbol{\chi}_i) - \hat{\mathbf{y}}_t)(h(\boldsymbol{\chi}_i) - \hat{\mathbf{y}}_t)^T + \mathbf{R}$
$\mathbf{P}_{\mathbf{x}\mathbf{y}_t} = \sum_{i=0}^{2n} W_i^{(c)} (f(\boldsymbol{\chi}_i) - \hat{\mathbf{x}}_{t|t-1})(h(\boldsymbol{\chi}_i) - \hat{\mathbf{y}}_t)^T$
$\mathbf{K}_t = \mathbf{P}_{\mathbf{x}\mathbf{y}_t} \mathbf{P}_{\mathbf{y}_t}^{-1}$
$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_m - \hat{\mathbf{y}}_t)$
$\mathbf{P}_{t|t} = \mathbf{P}_{t|t} - \mathbf{K}_t \mathbf{P}_{\mathbf{y}_t} \mathbf{K}_t^{-1}$

9: **end if**

10: **end for**

Outputs: $\{\hat{\mathbf{x}}_{t|t}\}_{t=0}^{t_{end}}$, $\{\mathbf{P}_{t|t}\}_{t=0}^{t_{end}}$

As for the observation, we have chosen a simple model where the body acquires some noisy measurement of its position (given for instance by a GPS):

$$\mathbf{y}_t = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_t + \mathbf{v}_t \tag{20}$$

with $\mathbf{v}_t \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R} = \sigma_v^2 \mathbf{I}_{2\times 2}\right)$

B. Framework

We consider a simple scenario where a circular trajectory (Fig. 2) is generated and corresponds to a motion of 20 seconds, beginning at $\mathbf{X}_0 = [0, 0, 0]^T$. The time update is performed at a sample rate of 100 Hz, hence $\Delta t = 0.01$ s and we consider the inputs $\mathbf{u}_t$ entering the model at the same rate. As for the measurements, they are delivered at a frequency of 10 Hz. For numerical stability purpose, we implemented the Square-Root version of the UKF [12].

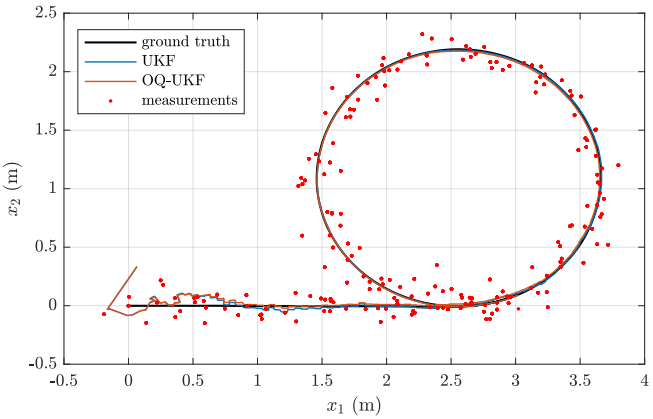


Fig. 2: Ground truth trajectory starting at $\mathbf{X}_0 = [0, 0, 0]^T$. Here, the uncertainties are parameterized as $\sigma_\theta = \frac{\pi}{180}$ rad/s, $\sigma_1 = \sigma_2 = 0.01$ m and $\sigma_v = 0.1$ m

The CLVQ procedure is applied as described in Algorithm 2. It is important to note that due to the fact that the system (19) has internal noises, the state had to be augmented with these noises. The CLVQ can either be applied on the full augmented state or just on specific state variables. In this study, we chose to apply it on the three states of interest (i.e. θ, x_1, x_2) and not the full augmented state. We would like to underline that we conducted experiment where we chose the state variable θ only, but we have noticed that in the case of the considered problem, the results were almost identical as if all three variables were chosen for the CLVQ.

To assess the improvement of the standard UKF using our method, we perform Monte-Carlo simulations. The metric used for validation is the Root-Mean-Square-Error (RMSE) of the estimation over the whole trajectory, averaged on all simulations.

C. Monte-Carlo simulations

During a given run, each estimator processes the same data to ensure a fair comparison. Besides, we always set the initial covariance to $\mathbf{P}_{0|0} = \text{diag}[(\frac{\pi}{6})^2, (0.3)^2, (0.3)^2]$ and the estimate is always initialized at the true state $\hat{\mathbf{X}}_{0|0} = \mathbf{X}_0 = [0, 0, 0]^T$. The true trajectory for each simulation is generated online by applying nominal inputs $\mathbf{u}_t$ disturbed by white Gaussian noise with covariance $\mathbf{Q}$. For the measurements, we also artificially add noise of the same level as that of the matrix $\mathbf{R}$. By doing so, we consider these matrices to be well-parameterized (i.e. the uncertainty we assume on the noise is the same as the true noise actually applied on the system). The tuning parameters of the CLVQ are $k_{max} = 100$, $\boldsymbol{\gamma}_k = (\frac{1}{20}\mathbf{S}_t)/k$, $\forall k = \overline{1, k_{max}}$ at the current time t.

We then run 400 Monte-Carlo simulations for different noises $\sigma_v^2 \in [10^{-4}, 9.10^{-2}]\,\text{m}^2$. In Fig. 3, we have displayed the Monte-Carlo average of the RMSE on $\{\theta_t\}_{t=0}^{t_{end}}$ on one side and the RMSE on $\{\mathbf{x}_t\}_{t=0}^{t_{end}}$ on the other side, as a function of σ_v^2. For both states, we compare the performance of the standard Kalman filter against the performance of the proposed approach. Results show that, for this trajectory and the problem (19-20) at stake, the OQ-UKF provides the best heading estimate as the noise level grows (Fig. 3a). As for the position estimate (Fig. 3b), if we zoom in a more precise scale, we can see that our method seems slightly better about one or two millimeters for high noises. The same conclusions can be drawn by fixing the measurement noise variance σ_v^2 to a given value and assessing the influence of $\mathbf{Q}$ on the error. Starting from a noise tuning $\mathbf{Q}_0 = \text{diag}[(\frac{\pi}{6})^2, (0.1)^2, (0.1)^2]$, we run 400 Monte-Carlo simulations with a given covariance $\mathbf{Q} = c\,\mathbf{Q}_0$ with a multiplier factor $c \in [1, 9]^T$. Results are displayed in Fig. 4.

D. Computational efficiency

In order to evaluate the computational effort, 100 simulations were performed, with a time update of 100 Hz and a measurement update of 10 Hz. Here, the CLVQ was performed on the θ variable only. Over these 100 simulations, the average computational time and their standard deviation was computed for the UKF and the OQ-UKF. The results are provided in Table II as well as their difference compared to the standard UKF given in terms of ratio $\frac{\text{[Filter] average time}}{\text{UKF average time}}$. We also added their respective performance by assessing the heading RMSE averaged on the 100 simulations.

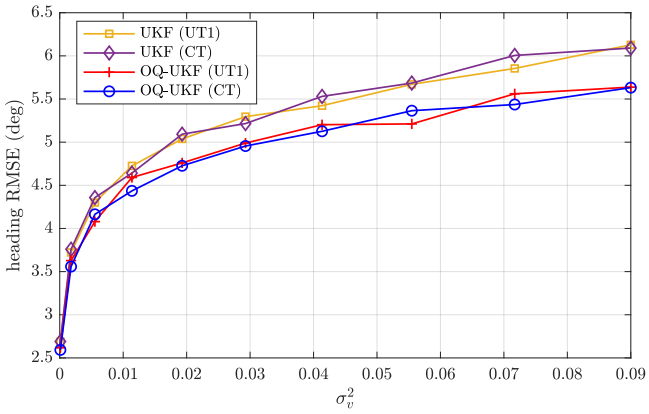
TABLE II: Average computational cost

Filter	average time (s)	time ratio	average RMSE θ (°)
UKF	0.27 ± 0.03	1	6.07 ± 1.35
OQ-UKF (k_{max}=300)	1.64 ± 0.14	≈6.1	5.23 ± 1.21
OQ-UKF (k_{max}=600)	3.17 ± 0.28	≈11.74	5.49 ± 1.30

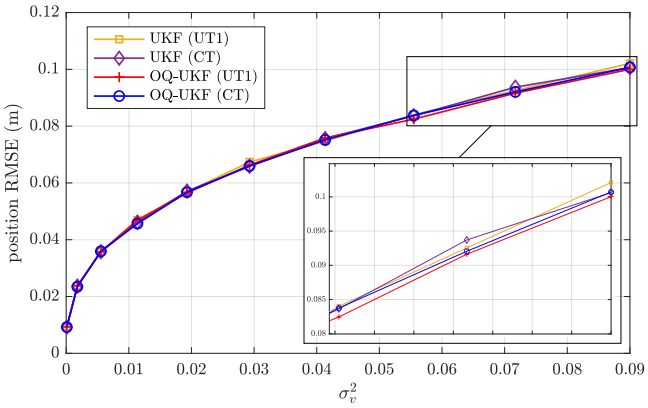
We can conclude that the proposed OQ-UKF is slower than the conventional UKF. The computational time is six times higher for $k_{max} = 300$ and almost as twice for $k_{max} = 600$ with a respective improvement of 13.84% and 9.56% on the average RMSE. As discussed in [15], the most time consuming procedure is the research of the winning index in Algorithm 1. In our paper, we did not focused on the optimization of the computational effort but future studies should involve this aspect.

VI. Conclusions and Discussions

In this work, we addressed the issues related to the approximations of the mean and the covariance of a random variable passing through a nonlinear function using the Unscented

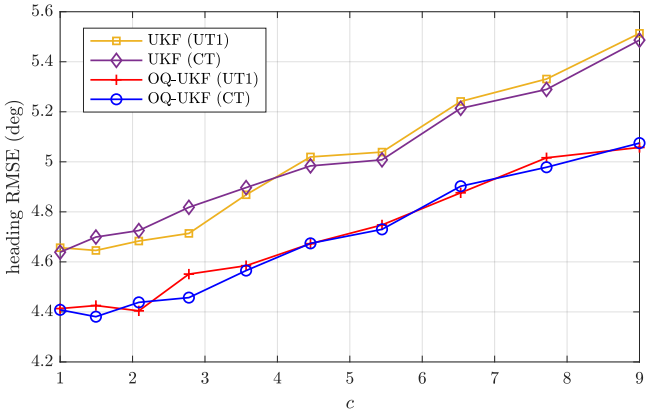


(a) Average RMSE on $\{\theta_t\}_{t=0}^{t_{end}}$ as a function of σ_v^2. For these simulations, we have fixed $\sigma_\theta = \frac{\pi}{6}$ rad/s, $\sigma_1 = \sigma_2 = 0.1$ m

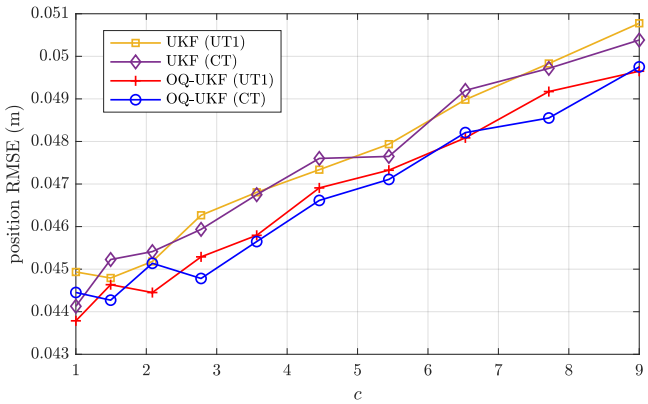


(b) Average RMSE on $\{\mathbf{x}_t\}_{t=0}^{t_{end}}$ as a function of σ_v^2. For these simulations, we have fixed $\sigma_\theta = \frac{\pi}{6}$ rad/s, $\sigma_1 = \sigma_2 = 0.1$ m

Fig. 3: Monte-Carlo average of the RMSE over the whole trajectory, as a function of the measurement noise σ_v^2



(a) Average RMSE on $\{\theta_t\}_{t=0}^{t_{end}}$ as a function of c. For these simulations, we have fixed $\sigma_v = 0.1$ m



(b) Average RMSE on $\{\mathbf{x}_t\}_{t=0}^{t_{end}}$ as a function of c. For these simulations, we have fixed $\sigma_v = 0.1$ m

Fig. 4: Monte-Carlo average of the RMSE over the whole trajectory, as a function of the model noise factor c.

Transform. We proposed a complementary approach, using Optimal quantization to approximate these moments. First, we illustrated this method with the example of a two-dimensional random vector transformed into a mono-dimensional random variable and showed that it can improve the performance of the UT. We then conducted simulations in the context of a mobile robot pose estimation using an UKF. Our results showed, in average, a reduction of the estimation error in this particular case.

However it should be pointed out that this method also needs further development to be fully reliable in a more general context. Hence, more theoretical work should be addressed in this direction. A direction of investigation could be the choice of the parameter k_{max} and the step sequence $\{\gamma_k\}_{k=1}^{k_{max}}$. This choice is already discussed in the literature [15] and we may not have taken the best values for our study. Besides, in this study we only considered the σ-points placement. Of course, since we have changed the points localization, it should also imply different weights. In fact, our first attempts to change the weights using the strategy given at the end of III-B resulted in failures when applied to the UKF. We think (but these are mere speculations) that using these new weights can't capture the moments accurately since there is not a sufficient number of σ-points to make it accurate. One could think that this method of weights computation is too "naive" given the fact that the UT uses a very limited number of points.

That being said, we believe that the Optimal quantization still provides a good support and has a lot of potential in order to deal with the problem of Bayesian estimation. For example, studies focused on the extension of the UT for circular variables under a Von Mises distribution [23], [24]. The Optimal quantization could be used to approach a Von Mises distribution, provided that one knows the parameters (mean and concentration coefficient) of this distribution. The extension to the Von Mises distribution could also be a line of investigation.

APPENDIX A

TABLE III: Three different sets of parameters for the UT

	α	β	κ
UT1 [1]	1	0	$3-n$
UT2 [3]	10^{-3}	2	0
CT [5]	1	0	0

In Table III, the UT1 set is equivalent to considering only the coefficient κ in the the UT since $\lambda = \alpha^2(n+\kappa) - n = \kappa$. This tuning completely coincides with Gauss-Hermite quadrature rule when $n = 1$. In the UT2 set, the coefficient $\alpha = 10^{-3}$ is used to gather the σ-points closer to the mean and $\beta = 2$ allows to take higher order moments into account. the CT set is equivalent to apply a no-scaling in the σ-points spreading. In [10], a more detailed explanation is provided concerning the influence of each parameter.

APPENDIX B

It would be interesting to see the evolution of (14) in the considered scenario of subsection III-C. We have displayed this evolution as a function of $\mathbf{P}$ in Fig. 5. As expected, the distortion has been reduced after the application of the CLVQ, at least for high covariances. Still, one can see that there is not a lot of difference, suggesting that the original placement is already not far from the optimality in the sense of the minimization of (14). But the fact that we were able to further reduce this quantity shows that the CLVQ algorithm works well and that the new placement is closer to the optimum than the old one. At this point, we still do not know if we have reached a local miminum or the global minimum. In any case, achieving the perfect minimum is not our goal. Recall that the purpose of our method is to do a compromise between Gauss quadrature approach and Optimal quantization approach. Completely embrace the Optimal quantization approach would probably deteriorate the Kalman Filter since we could lose some properties that are necessary in the UKF.

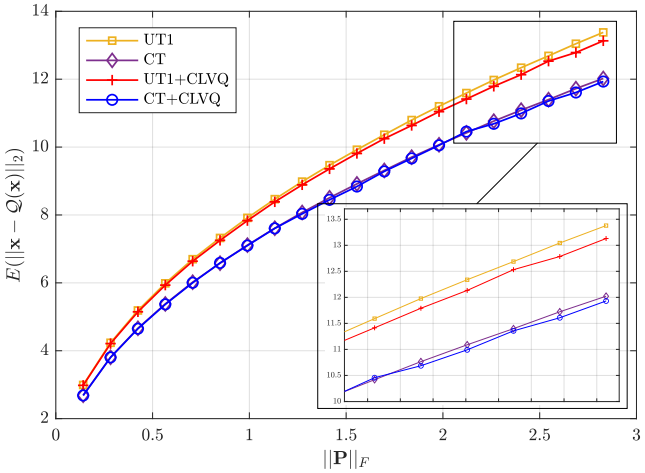


Fig. 5: Distortion in function of $\mathbf{P}$

REFERENCES

[1] Julier, S. J., Uhlmann, J. K., Durrant-Whyte, H. F. (1995, June). A new approach for filtering nonlinear systems. In Proceedings of 1995 American Control Conference-ACC'95 (Vol. 3, pp. 1628-1632). IEEE.

[2] Roth, M., Hendeby, G., Gustafsson, F. (2016). Nonlinear Kalman filters explained: A tutorial on moment computations and sigma point methods. Journal of Advances in Information Fusion, 11(1), 47-70.

[3] Wan, E. A., Van Der Merwe, R. (2000, October). The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373) (pp. 153-158). Ieee.

[4] Julier, S. J. (2002, May). The scaled unscented transformation. In Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301) (Vol. 6, pp. 4555-4559). IEEE.

[5] Arasaratnam, I., Haykin, S. (2009). Cubature kalman filters. IEEE Transactions on automatic control, 54(6), 1254-1269.

[6] Straka, O., Duník, J., Šimandl, M. (2014). Unscented Kalman filter with advanced adaptation of scaling parameter. Automatica, 50(10), 2657-2664.

[7] Scardua, L. A., Da Cruz, J. J. (2017). Complete offline tuning of the unscented Kalman filter. Automatica, 80, 54-61.

[8] Sakai, A., Kuroda, Y. (2010). Discriminative parameter training of unscented Kalman filter. IFAC Proceedings Volumes, 43(18), 677-682.

[9] Masoumnezhad, M., Jamali, A., Nariman-zadeh, N. (2015). Optimal design of symmetrical/asymmetrical sigma-point Kalman filter using genetic algorithms. Transactions of the Institute of Measurement and Control, 37(3), 425-432.

[10] Nielsen, K., Svahn, C., Rodriguez-Deniz, H., Hendeby, G. (2021, September). Ukf parameter tuning for local variation smoothing. In 2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI) (pp. 1-8). IEEE.

[11] Rhudy, M. (2023). Selection of Tuning Parameters of the Unscented Kalman Filter using Analytical Truth Statistics. In AIAA SCITECH 2023 Forum (p. 2702).

[12] Van Der Merwe, R., Wan, E. A. (2001, May). The square-root unscented Kalman filter for state and parameter-estimation. In 2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221) (Vol. 6, pp. 3461-3464). IEEE.

[13] Graf, S., Luschgy, H. (2007). Foundations of quantization for probability distributions. Springer.

[14] Pagès, G., Pham, H. (2005). Optimal quantization methods for nonlinear filtering with discrete-time observations. Bernoulli, 11(5), 893-932.

[15] Pagès, G., Printems, J. (2003). Optimal quadratic quantization for numerics: the Gaussian case. Monte Carlo Methods Appl., 9(2), 135-165.

[16] Pagès, G., Pham, H., Printems, J. (2004). Optimal quantization methods and applications to numerical problems in finance. Handbook of computational and numerical methods in finance, 253-297.

[17] Sellami, A. (2005). Méthodes de quantification optimale pour le filtrage et applications à la finance (Doctoral dissertation, Université Paris Dauphine-Paris IX).

[18] Steinbring, J., Pander, M., Hanebeck, U. D. (2015). The smart sampling Kalman filter with symmetric samples. arXiv preprint arXiv:1506.03254.

[19] Hanebeck, U. D., Huber, M. F., Klumpp, V. (2009, December). Dirac mixture approximation of multivariate gaussian densities. In Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference (pp. 3851-3858). IEEE.

[20] Arasaratnam, I., Haykin, S., Elliott, R. J. (2007). Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature. Proceedings of the IEEE, 95(5), 953-977.

[21] Štecha, J., Havlena, V. (2012, July). Unscented kalman filter revisited—Hermite-Gauss quadrature approach. In 2012 15th International Conference on Information Fusion (pp. 495-502). IEEE.

[22] Ito, K., Xiong, K. (2000). Gaussian filters for nonlinear filtering problems. IEEE transactions on automatic control, 45(5), 910-927.

[23] Chen, M. Y., Wang, H. Y. (2016). Nonlinear measurement update for recursive filtering based on the gauss von mises distribution. Procedia Computer Science, 92, 543-548.

[24] Kurz, G., Gilitschenski, I., Hanebeck, U. D. (2016). Unscented von mises–fisher filtering. IEEE Signal Processing Letters, 23(4), 463-467.