Comparison of Filtering Algorithms in Vehicle Positioning by Using Low-Cost Sensors

Trung Dung Pham, Volker Schwieger
Institute of Engineering Geodesy, University of Stuttgart, Geschwister-Scholl-Str. 24D, 70174 Stuttgart, Germany
dung.pham@ingeo.uni-stuttgart.de, volker.schwieger@ingeo.uni-stuttgart.de

Keywords: Extended Kalman filter, Unscented Kalman filter, Particle filter, Low-cost sensors

Abstract: To extend the aim of vehicle positioning using data of low-cost sensors, several filtering algorithms including Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filter (PF) are studied and compared. These algorithms are compared in a highly non-linear prediction model and measurement model. The measurement noise is simulated by Gaussian and non-Gaussian distributions. The root mean square error is used as accuracy parameter. It is found that the PF is about five times and nine times more accurate than the UKF and the EKF, respectively. Although the accuracy of both the UKF and the EKF reduces approximately 1.5 times with using non-Gaussian distributed noise, the accuracy of the PF is not influenced by the probability distribution of the noise. In terms of computational time the EKF is the best solution followed by the UKF while the PF is the worst case.

1 INTRODUCTION

The estimation methods for non-linear systems widely used in vehicle navigation have been studied for many years. In 1960, Kalman proposed a well-known algorithm called Kalman filter (Kalman, 1960). The filter is applicable for linear models containing Gaussian noise. However, almost all real systems are characterized by non-linear phenomena; therefore, the KF technique cannot be used directly. To overcome this obstacle, Welch and Bishop (1995) developed a new method named Extended Kalman Filter by linearizing the non-linear systems at an operational point to achieve the linear models. But the disadvantage of the EKF is the divergence phenomenon in case of the highly non-linear model. A better method for dealing with the non-linear model without the linearization step is proposed by Julier and Uhlman (1997), named the Unscented Kalman Filter. Unfortunately, both the EKF and the UKF are only suitable for Gaussian noise. Therefore, the Particle Filter method is proposed to handle systems in case of measurements with non-Gaussian noise. This method is well-known as Bootstrap method (Gordon, 1993) and Sequence Monte Carlo method (Liu and Chen, 1998).

Low-cost sensors have been installed in many applications of engineering navigation and geodesy for several years. Ramm (2008) developed three approaches for the KF for vehicle positioning in which the observation model includes the measurement of Global Positioning System (GPS) and of several low-cost sensors. Besides, Alkhatib (2008) introduced a comparison of three algorithms, the EKF, the UKF, and the PF for the non-linear state estimation. Here, the observations of low-cost sensors are simulated. In addition, Schweitzer (2012) compared the accuracy of position between two prediction models which are the straight line model and the circle prediction model. Both are integrated into the KF algorithm. In this case, the observations, using data measured by several vehicle sensors and GPS receivers, are linearly related to the introduced state parameters. Aussems (1999) also described a circle model using for tracking a vehicle trajectory. In the study, the coordinates of the vehicle are non-linearly related to the state parameters including the velocity, the angular velocity and the pitch angle. Several researches presented solutions for non-linear observation systems in navigation applications. For example, Sternberg (2000) demonstrated an observation system which is non-linearly related to its state parameters. In this case, the measurement consists of angles, distances, and point coordinates. Additionally, Julier and Uhlman (2004) invented an

alternative solution by using the unscented transform method to replace the KF according to the non-linear system. Moreover, the EKF, the UKF and the PF method used for this issue can be also found in Särkkä (2013).

In this paper, the non-linear relationship will be not only assumed for the prediction model, but also for the observation model. This issue dealt with non-linear observation models by Sternberg (2000), Julier and Uhlman (2004), and Särkkä (2013). On the other hand, it includes the investigation of the Gaussian distribution of measurement noise carried out by Alkhatib (2008) as well as Schweitzer (2012), so these situations will be expanded for both cases (Gaussian noise and non-Gaussian noise).

Section 2 outlines the EKF, the UKF and the PF for non-linear systems. Section 3 investigates an approach for the vehicle positioning by the kinematic non-linear prediction model and the non-linear observation model. In section 4, the accuracy of position and the computational time are analyzed and compared by means of a numerical experiment.

2 MATHEMATICAL TECHNIQUES

2.1 EKF Algorithm

The Extended Kalman Filter method is used for a prediction process and a measurement process which are described by non-linear functions (Welch and Bishop, 1995):

$$\boldsymbol{y}_{k+1} = \phi(\boldsymbol{y}_k, \boldsymbol{u}_k, \boldsymbol{w}_k), \tag{1}$$

$$\boldsymbol{l}_{k+1} = \varphi(\boldsymbol{y}_{k+1}, \boldsymbol{v}_{k+1}), \tag{2}$$

where $\boldsymbol{l}_{k+1}$ is the vector of measurement, $\boldsymbol{y}_{k+1}$ is the state vector at epoch *k+1*, $\boldsymbol{u}_k$ is the vector of acting forces, and $\boldsymbol{w}_k$, $\boldsymbol{v}_{k+1}$ are the process and measurement noise which are uncorrelated Gaussian noise. ϕ, φ are the non-linear process and measurement functions, respectively. The EKF algorithm is performed in two steps:

1. Prediction step

The prediction of the state vector is expressed by:

$$\bar{\boldsymbol{y}}_{k+1} = \phi(\hat{\boldsymbol{y}}_k). \tag{3}$$

The covariance matrix of the predicted state vectors is computed in equation (4) (without acting forces):

$$\boldsymbol{\Sigma}_{\bar{y}_{k+1}\bar{y}_{k+1}} = \boldsymbol{T}_k \boldsymbol{\Sigma}_{\hat{y}_k\hat{y}_k} \boldsymbol{T}_k^T + \boldsymbol{S}_k \boldsymbol{\Sigma}_{\boldsymbol{w}_k\boldsymbol{w}_k} \boldsymbol{S}_k^T, \tag{4}$$

by inserting the transition matrix $\boldsymbol{T}_k = \left(\frac{\partial\phi(\boldsymbol{y}_k,\boldsymbol{u}_k,\boldsymbol{w}_k)}{\partial \boldsymbol{y}_k}\right)$, the matrix of distubance quantities $\boldsymbol{S}_k = \left(\frac{\partial\phi(\boldsymbol{y}_k,\boldsymbol{u}_k,\boldsymbol{w}_k)}{\partial \boldsymbol{w}_k}\right)$ and covariance matrices of the process noise $\boldsymbol{\Sigma}_{\boldsymbol{w}_k\boldsymbol{w}_k}$.

2. Update step

The transformation of the predicted state vector into the observation space

$$\bar{\boldsymbol{l}}_{k+1} = \varphi(\bar{\boldsymbol{y}}_{k+1}) \tag{5}$$

is used to calculate the vector of innovations

$$\boldsymbol{d}_{k+1} = \boldsymbol{l}_{k+1} - \bar{\boldsymbol{l}}_{k+1}. \tag{6}$$

To compute the corresponding covariance matrix of innovations equation (6) must be linearized. So the design matrix $\boldsymbol{A}_{k+1} = \left(\frac{\partial\varphi(\boldsymbol{y}_{k+1},\boldsymbol{v}_{k+1})}{\partial \boldsymbol{y}_{k+1}}\right)$ is determined and one gets:

$$\boldsymbol{\Sigma}_{\boldsymbol{d}_{k+1}\boldsymbol{d}_{k+1}} = \boldsymbol{A}_{k+1} \boldsymbol{\Sigma}_{\bar{y}_{k+1}\bar{y}_{k+1}} \boldsymbol{A}_{k+1}^T + \boldsymbol{\Sigma}_{\boldsymbol{l}_{k+1}\boldsymbol{l}_{k+1}}, \tag{7}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{l}_{k+1}\boldsymbol{l}_{k+1}}$ is the covariance matrix of the measurement.

The update value and the covariance matrix of the state vector are computed by the following equations:

$$\hat{\boldsymbol{y}}_{k+1} = \bar{\boldsymbol{y}}_{k+1} + \boldsymbol{K}_{k+1} \boldsymbol{d}_{k+1}, \tag{8}$$

$$\boldsymbol{\Sigma}_{\hat{y}_{k+1}\hat{y}_{k+1}} = \boldsymbol{\Sigma}_{\bar{y}_{k+1}\bar{y}_{k+1}} - \boldsymbol{K}_{k+1} \boldsymbol{\Sigma}_{\boldsymbol{d}_{k+1}\boldsymbol{d}_{k+1}} \boldsymbol{K}_{k+1}^T, \tag{9}$$

by inserting the Kalman gain matrix

$$\boldsymbol{K}_{k+1} = \boldsymbol{\Sigma}_{\bar{y}_{k+1}\bar{y}_{k+1}} \boldsymbol{A}_{k+1}^T \boldsymbol{\Sigma}_{\boldsymbol{d}_{k+1}\boldsymbol{d}_{k+1}}^{-1}. \tag{10}$$

2.2 UKF Algorithm

The Unscented Kalman Filter equations, presented in Julier and Uhlman (1997), can be applied for the two processes shown in equation (1) and (2). The

UKF algorithm can be described by the following procedure:

1. *Generation of the sigma-points and the corresponding weights*

Generation of (*2n+1*) sigma-points by using the state vector and its covariance matrix is given by:

$$\boldsymbol{\gamma}_k = \left[\hat{\boldsymbol{y}}_k \quad \hat{\boldsymbol{y}}_k \pm \sqrt{n+\lambda} \cdot \sqrt{\boldsymbol{\Sigma}_{\hat{y}_k \hat{y}_k}}\right], \qquad (11)$$

where *n* is the dimension of the updated state vector $\hat{\boldsymbol{y}}_k$ at the epoch *k* and $\boldsymbol{\Sigma}_{\hat{y}_k \hat{y}_k}$is covariance matrix of the updated state vector.
Weighted sigma-points are defined as:

$$W_0^{(m)} = \lambda/(n+\lambda), \qquad (12)$$

$$W_0^{(c)} = \lambda/(n+\lambda) + (1-\alpha^2+\beta), \qquad (13)$$

$$W_i^{(m)} = W_i^{(c)} = 1/2(n+\lambda) \;\; i = 1,\ldots,2n, \qquad (14)$$

in which the parameters are determined by the following method:

The first parameter α is set between 10^{-4} and 1 determining the spread of the sigma-points around the mean value. The secondary scaling parameter κ is usually set to *3-n*. For a Gaussian distribution the third parameter β is 2 (Wan and Van der Merwe, 2002). $\sqrt{n+\lambda}$ is a scalar scaling factor and λ is determined by:

$$\lambda = \alpha^2(n+\kappa) - n. \qquad (15)$$

2. Prediction step

Each sigma-point passes through the non-linear prediction function ϕ, as shown in (16):

$$\boldsymbol{\chi}_k^{(i)} = \phi\left(\boldsymbol{\gamma}_k^{(i)}\right), i = 0,1,2,\ldots,2n. \qquad (16)$$

The mean value and the covariance matrix of the state vector can be expressed by using these sigma-points after transforming them by the following equations (without acting forces):

$$\bar{\boldsymbol{y}}_{k+1} = \sum_{i=0}^{2n} W_i^{(m)} \boldsymbol{\chi}_k^{(i)}, \qquad (17)$$

$$\boldsymbol{\Sigma}_{\bar{y}_{k+1}\bar{y}_{k+1}} = \boldsymbol{S}_k \boldsymbol{\Sigma}_{\boldsymbol{w}_k \boldsymbol{w}_k} \boldsymbol{S}_k^T + \sum_{i=0}^{2n} W_i^{(c)} (\boldsymbol{\chi}_k^{(i)} - \bar{\boldsymbol{y}}_{k+1})(\boldsymbol{\chi}_k^{(i)} - \bar{\boldsymbol{y}}_{k+1})^T. \qquad (18)$$

3. Update step

The sigma-points are transmited through the non-linear measurement function φ

$$\boldsymbol{\psi}_{k+1}^{(i)} = \varphi\left(\boldsymbol{\chi}_{k+1}^{(i)}\right), i = 0,1,2,\ldots,2n \qquad (19)$$

where $\boldsymbol{\psi}$ is a matrix of the output sigma-points.
The predicted observations are calculated by:

$$\bar{\boldsymbol{l}}_{k+1} = \sum_{i=0}^{2n} W_i^{(m)} \cdot \boldsymbol{\psi}_{k+1}^{(i)} \qquad (20)$$

Julier and Uhlman (1997) described how the innovation covariance matrix is determined by adding the covariance of measurement noise $\boldsymbol{\Sigma}_{\boldsymbol{l}_{k+1}\boldsymbol{l}_{k+1}}$ at epoch *k+1* and the covariance matrix of the posterior sigma points as follows:

$$\boldsymbol{\Sigma}_{\bar{l}_{k+1}\bar{l}_{k+1}} = \boldsymbol{\Sigma}_{\boldsymbol{l}_{k+1}\boldsymbol{l}_{k+1}} + \sum_{i=0}^{2n} W_i^{(c)} \left(\boldsymbol{\psi}_{k+1}^{(i)} - \bar{\boldsymbol{l}}_{k+1}\right)\left(\boldsymbol{\psi}_{k+1}^{(i)} - \bar{\boldsymbol{l}}_{k+1}\right)^T \qquad (21)$$

According to the Kalman filter, the Kalman gain matrix $\boldsymbol{K}$ is defined by equation 10.
Finally, the updated state vector and the updated covariance matrix of the state vector are computed by equation 8 and equation 9.

2.3 Particle Filter

The Particle filter (PF) is based on the Monte Carlo (MC) technique to estimate state variables. The posterior density function of the state vector is determined by optimal recursive Bayesian estimation using the available prior knowledge. There were several versions of PF developed over the last few decades. Gordon et al. (1993) proposed a Sampling Importance Resampling (SIR) method. This method is closely related to the Bootstrap procedure which was introduced by Efron in the late 1970s (Efron, 1979). Pitt and Shephard (1999) introduced the Auxiliary Particle filter dealing with tailed measurement densities to enhance some limitations of SIR. Furthermore, Doucet et al. (2000) developed a novel extension of the important sampling technique named the Sequential Importance Sampling (SIS). The PF theory based on the SIR algorithm that is used as numerical example in section 4 will be described below.

A prediction probability density function $p(\boldsymbol{y}_{k+1}|\boldsymbol{l}_{1:k})$ can be determined by using a probability model of state evolution $p(\boldsymbol{y}_{k+1}|\boldsymbol{y}_k)$ which is described in equation (1) as follows:

$$p(\boldsymbol{y}_{k+1}|\boldsymbol{l}_{1:k}) = \int p(\boldsymbol{y}_{k+1}|\boldsymbol{y}_k)p(\boldsymbol{y}_k|\boldsymbol{l}_{1:k})d\boldsymbol{y}_k \qquad (22)$$

where $\boldsymbol{l}_{1:k} = \{l_1, l_2, \dots, l_k\}$ and a filtering probability density $p(\boldsymbol{y}_k|\boldsymbol{l}_{1:k})$ for the Bayesian inference is given by:

$$p(\boldsymbol{y}_k|\boldsymbol{l}_{1:k}) = \frac{p(\boldsymbol{l}_k|\boldsymbol{y}_k)p(\boldsymbol{y}_k|\boldsymbol{l}_{1:k-1})}{p(\boldsymbol{l}_k|\boldsymbol{l}_{1:k-1})} \qquad (23)$$

The PF approximates the probability density function $p(\boldsymbol{y}_k|\boldsymbol{l}_{1:k})$ by a significant number of *N* independent particles $\left\{\boldsymbol{y}_k^{(i)}\right\}_{i=1}^{N}$ and their associated weights $\left\{w_k^{(i)}\right\}_{i=1}^{N}$, where the sum of all weights is proportional to the unity. The PF updates the state vector and the corresponding weights recursively with each new measurement. In equation (23), the normalization factor $p(\boldsymbol{l}_k|\boldsymbol{l}_{1:k-1})$ is usually unknown. But this factor is not essential for this method, since the probability density function $p(\boldsymbol{y}_k|\boldsymbol{l}_{1:k})$ can be sufficiently evaluated by:

$$p(\boldsymbol{y}_{k+1}|\boldsymbol{l}_{1:k}) \propto p(\boldsymbol{l}_k|\boldsymbol{y}_k) \cdot p(\boldsymbol{y}_k|\boldsymbol{l}_{1:k-1}), \qquad (24)$$

where, under assumption of Gaussian distribution, the likelihood function $p(\boldsymbol{l}_k|\boldsymbol{y}_k)$ is computed by considering the measurement standard deviation (STD) σ

$$p(\boldsymbol{l}_k|\boldsymbol{y}_k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(l_k - y_k)^2}{2\sigma^2}} \qquad (25)$$

and $p(\boldsymbol{y}_k|\boldsymbol{l}_{1:k-1})$ is approximated with particles which are known as the main idea of the PF method, according to

$$p(\boldsymbol{y}_k|\boldsymbol{l}_{1:k-1}) \approx \frac{1}{N}\sum_{i=1}^{N} \delta\left(\boldsymbol{y}_k - \boldsymbol{y}_k^{(i)}\right), \qquad (26)$$

where $\delta(\cdot)$ is the delta-Dirac function.

However, this approach suffers from divergence phenomena that almost all the particles' weights have a value of zero except one non-zero weight after a few steps. This problem can be handled by a resampling step. Several resampling algorithms were analyzed and compared by Douc et al. (2005). The SIR algorithm or the Bootstrap algorithm is briefly described by the following procedure:

1. *N* particles $\{\boldsymbol{y}_0\}_{i=1}^{N}$ are drawn depending on the Gaussian distribution $p(\boldsymbol{y}_0)$ at epoch *k=0*.
2. The weights $w_k^{(i)} = p\left(\boldsymbol{l}_k|\boldsymbol{y}_k^{(i)}\right)$ are calculated by equation (25) and normalised weights can be defined

$$\widetilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}, i = 1,2, \dots, N. \qquad (27)$$

3. A new set of particles $\left\{\acute{\boldsymbol{y}}_k^{(i)}\right\}_{i=1}^{N}$ are rearranged from the current set $\left\{\boldsymbol{y}_k^{(i)}\right\}_{i=1}^{N}$ by the resampling method, where $prob\left(\acute{\boldsymbol{y}}_k^{(i)} = \boldsymbol{y}_k^{(j)}\right) = \widetilde{w}_k^{(j)}$ (see figure 1).
4. At epoch *k+1*, new particles $\left\{\boldsymbol{y}_{k+1}^{(i)}\right\}_{i=1}^{N}$ are drawn which are based on the prediction function (equation 1) as follows:

$$\boldsymbol{y}_{k+1}^{(i)} = \phi\left(\acute{\boldsymbol{y}}_k^{(i)}, \boldsymbol{u}_k^{(i)}, \boldsymbol{w}_k^{(i)}\right), i = 1,2, \dots, N$$

where the process noise $\boldsymbol{w}_k^{(i)}$ is simulated by Gaussian distribution and the acting force $\boldsymbol{u}_k^{(i)}$is ignored.
5. Increase epoch *k:=k+1* and repeat from step 2.

The update state vector and the covariance of the state vector of the SIR algorithm are computed by:

$$\begin{aligned} \hat{\boldsymbol{y}}_k &= E(\boldsymbol{y}_k|\boldsymbol{l}_{1:k}) \\ &= \int \boldsymbol{y}_k p(\boldsymbol{y}_k|\boldsymbol{l}_{1:k})d\boldsymbol{y}_k \approx \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{y}_k^{(i)} \end{aligned} \qquad (28)$$

$$\begin{aligned} \boldsymbol{\Sigma}_{\hat{\boldsymbol{y}}_k\hat{\boldsymbol{y}}_k} &= \int (\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k)(\boldsymbol{y}_k \\ &\qquad - \hat{\boldsymbol{y}}_k)^T p(\boldsymbol{y}_k|\boldsymbol{l}_{1:k})d\boldsymbol{y}_k \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \left(\boldsymbol{y}_k^{(i)} - \hat{\boldsymbol{y}}_k\right)\left(\boldsymbol{y}_k^{(i)} - \hat{\boldsymbol{y}}_k\right)^T. \end{aligned} \qquad (29)$$

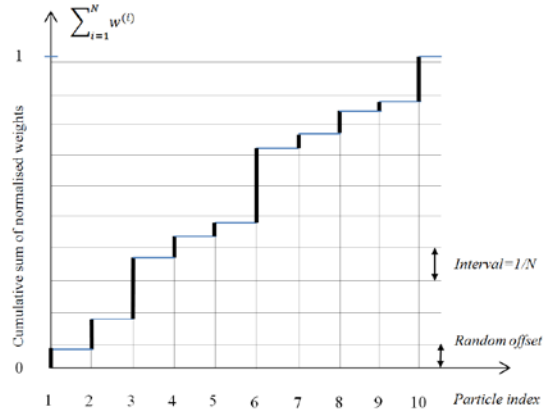

Figure 1: A graphical explanation of the systematic resampling method with ten particles (N=10). The random offset is chosen randomly from a uniform distribution over $(0,1/N]$ and the interval is proportional to $1/N$. The vertical axis of the graph illustrates the weight $w^{(i)}$ which is increased by the cumulative sum of the normalized weights $w^{(i)} = \sum_{j=1}^{i} w^{(j)}$. For this situation, illustrated in this figure, particle number 6 is chosen three times, particle number 3 and 10 are chosen twice, particle number 2, 4 and 8 are chosen once. Thus, the resampled set consists of particle indexes 2, 3, 3, 4, 6, 6, 6, 8, 10, 10.

3 APPROACHES FOR POSITION ESTIMATION OF VEHICLES

3.1 The Kinematic Prediction Model

This scenario illustrates a vehicle moving along a straight line trajectory. The corresponding prediction model has been developed by the Institute of Engineering Geodesy (IIGS) at the University of Stuttgart (figure 2). The state vector including the current horizontal coordinates x and y, the velocity v, the orientation φ, and the angular rotation $\Delta\varphi$ are defined in Schweitzer (2012):

$$\boldsymbol{y} = |x \quad y \quad \varphi \quad v \quad \Delta\varphi|^T, \tag{30}$$

Assume that, firstly, the vehicle moves on a straight line between two sequent measuring points, and secondly, the prediction is an approximation from one measuring epoch to the next measuring epoch. Figure 2 shows how predicted coordinates can be determined by combining the coordinates of the previous epoch and the current angular orientation in a polar survey method. Thus the prediction of the state vector in equation (1) is realized by the following equations:

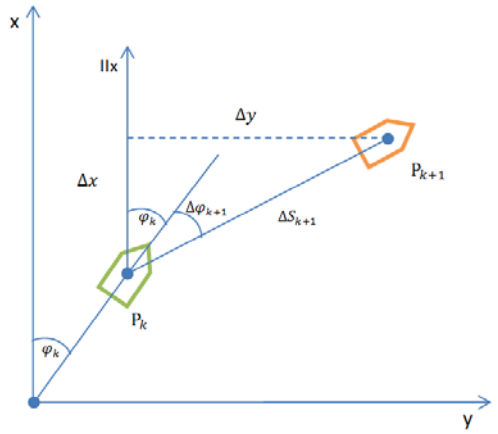

Figure 2: Straight line kinematic model

$$\begin{aligned}
\bar{x}_{k+1} &= \hat{x}_k + \Delta S_{k+1} \cdot cos(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1}) \\
&= \hat{x}_k + \bar{v}_{k+1} \cdot \Delta t \cdot cos(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1}), \\
\bar{y}_{k+1} &= \hat{y}_k + \Delta S_{k+1} \cdot sin(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1}) \\
&= \hat{y}_k + \bar{v}_{k+1} \cdot \Delta t \cdot sin(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1}), \\
\bar{\varphi}_{k+1} &= \hat{\varphi}_k + \Delta\bar{\varphi}_{k+1}, \\
\bar{v}_{k+1} &= \hat{v}_k, \\
\Delta\bar{\varphi}_{k+1} &= \Delta\bar{\varphi}_k.
\end{aligned} \tag{31}$$

The transition matrix $\boldsymbol{T}$ is expressed:

$$\boldsymbol{T}_k = \begin{vmatrix} 1 & 0 & T_{1,3} & T_{1,4} & T_{1,5} \\ 0 & 1 & T_{2,3} & T_{2,4} & T_{2,5} \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}, \tag{32}$$

with $T_{1,3} = -\bar{v}_{k+1} \cdot \Delta t \cdot sin(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1})$,
$T_{1,4} = \Delta t \cdot cos(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1})$,
$T_{1,5} = -\bar{v}_{k+1} \cdot \Delta t \cdot sin(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1})$,
$T_{2,3} = \bar{v}_{k+1} \cdot \Delta t \cdot cos(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1})$,
$T_{2,4} = \Delta t \cdot sin(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1})$,
$T_{2,5} = \bar{v}_{k+1} \cdot \Delta t \cdot sin(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1})$

where Δt is time interval and the disturbance matrix $\boldsymbol{S}$ is given by:

$$\boldsymbol{S}_k^T = \begin{vmatrix} S_{1,1} & S_{2,1} & 0 & \Delta t & 0 \\ S_{1,2} & S_{2,2} & \Delta t & 0 & \Delta t \end{vmatrix}, \tag{33}$$

with $S_{1,1} = \frac{\Delta t^2}{2} cos(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1})$,
$S_{1,2} = -\bar{v}_{k+1} \cdot \Delta t^2 \cdot \sin(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1})$,
$S_{2,1} = \frac{\Delta t^2}{2} cos(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1})$,
$S_{2,2} = \bar{v}_{k+1} \cdot \Delta t^2 \cdot \cos(\hat{\varphi}_k + \Delta\bar{\varphi}_{k+1})$,

The process noise covariance matrix $\boldsymbol{\Sigma}_{ww}$ is:

$$\boldsymbol{\Sigma}_{w_k w_k} = \begin{vmatrix} \sigma_{a_w}^2 & 0 \\ 0 & \sigma_{\varphi_w}^2 \end{vmatrix}, \tag{34}$$

where $\sigma_{a_w}, \sigma_{\varphi_w}$ are the STD of disturbance acceleration and disturbance rotational rate, respectively.

3.2 The Measurement Model

By way of example in this paper a simple observation vector is introduced. This observation vector comprises distances and bearings

$$\boldsymbol{l}_{k+1} = \left| s_{k+1}^{(1)} \quad s_{k+1}^{(2)} \quad \varphi_{k+1}^{(1)} \quad \varphi_{k+1}^{(2)} \right|^T. \tag{35}$$

In the corresponding measurement model, $s_{k+1}^{(i)}, \varphi_{k+1}^{(i)}$ are functions of the horizontal coordinates determined by the following equations:

$$\begin{aligned} s_{k+1}^{(i)} &= \sqrt{\Delta x_{k+1}^2 + \Delta y_{k+1}^2}, \\ \varphi_{k+1}^{(i)} &= atan\left(\frac{\Delta y_{k+1}}{\Delta x_{k+1}}\right). \end{aligned} \tag{36}$$

Hence, the components of the design matrix $\boldsymbol{A}$ are:

$$\boldsymbol{A}_{k+1} = \begin{vmatrix} \frac{\Delta x_{k+1}}{S} & \frac{\Delta y_{k+1}}{S} & 0 & 0 & 0 \\ \frac{\Delta x_{k+1}}{S} & \frac{\Delta y_{k+1}}{S} & 0 & 0 & 0 \\ -\frac{\Delta y_{k+1}}{S^2} & \frac{\Delta x_{k+1}}{S^2} & 0 & 0 & 0 \\ -\frac{\Delta y_{k+1}}{S^2} & \frac{\Delta x_{k+1}}{S^2} & 0 & 0 & 0 \end{vmatrix}, \tag{37}$$

where $\Delta x_{k+1} = x_{k+1} - X_0^{(i)}$, $\Delta y_{k+1} = y_{k+1} - Y_0^{(i)}$ and $\left(X_0^{(i)}, Y_0^{(i)}\right)$ are the known horizontal coordinates of the sensors with $i = \{1,2\}$ (see figure 3). Finally, the measurement noise covariance matrix is defined:

$$\boldsymbol{\Sigma}_{\boldsymbol{l}_{k+1}\boldsymbol{l}_{k+1}} = \begin{vmatrix} \sigma_S^2 & 0 & 0 & 0 \\ 0 & \sigma_S^2 & 0 & 0 \\ 0 & 0 & \sigma_\varphi^2 & 0 \\ 0 & 0 & 0 & \sigma_\varphi^2 \end{vmatrix}, \tag{38}$$

where σ_s and σ_φ are the STD of distances and bearings with the assumption that $\sigma_{S_1} = \sigma_{S_2} = \sigma_s$, and $\sigma_{\varphi_1} = \sigma_{\varphi_2} = \sigma_\varphi$.

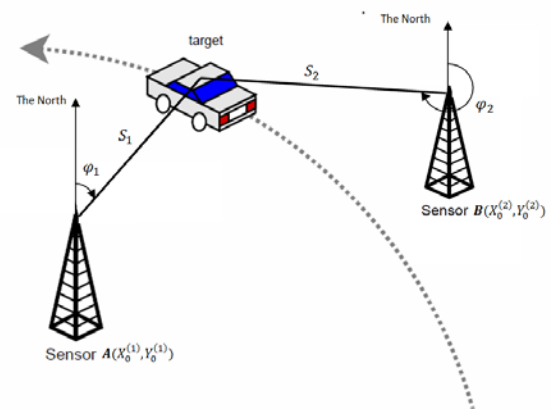

Figure 3: A car runs in a curve line trajectory which is considered as a combination of many sequent straight lines. Each straight line is created by one measuring epoch to the next one. In this figure, the motion of a moving car at epoch *k+1* is measured by two sensors which are located in reference points ***A*** and ***B***. They are able to measure the distances $s_{k+1}^{(1)}, s_{k+1}^{(2)}$ and the bearings $\varphi_{k+1}^{(1)}, \varphi_{k+1}^{(2)}$.

4 NUMERICAL APPLICATIONS

It is the aim of this section to compare the performance of different filtering algorithms which are presented in section 2 in terms of estimated accuracy and computational time.

4.1 Comparison of Estimated Accuracy

Root Mean Square Error (RMSE) of horizontal coordinates is used as accuracy parameter of the filtering algorithms. The RMSE computed for all measuring epochs is defined by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}[\Delta x^2 + \Delta y^2]}{m}}, \tag{39}$$

with $\Delta x = x_{true}^{(i)} - x_{est}^{(i)}$,
$\Delta y = y_{true}^{(i)} - y_{est}^{(i)}$,

where *m* is the number of measuring epochs, x_{true}, y_{true} are horizontal coordinates of the true trajectory, x_{est}, y_{est} are horizontal coordinates of the estimated trajectory.

In general, the measurements of low-cost sensors contain a high value of variance as well as unknown distributed noise. Besides, in reality the vehicle moves on a non-linear trajectory. Thus, the analysis of the three filtering algorithms for non-linear

models with respect to the high measurement variance is necessary. The measurement variances are simulated by increasing the STD of the distance and fixing the STD of the bearing. As an example, the results for three cases are shown in the following: (i) the measurements are affected by a small measurement noise of 0.1 m for distance and of 0.02 rad for bearing; (ii) those of medium noise behavior contain STDs of 0.5 m for distance and of 0.02 rad for bearing; and (iii) those of high noise behavior consist of STDs of 1.0 m for distance and of 0.02 rad for bearing. On the other hand, the measurement noise is simulated based on Gaussian distribution and non-Gaussian distribution (triangular distribution). The following estimated results of the PF, the UKF and the EKF are carried out for 100 measuring epochs with a time interval of 0.1 second within 1000 particles of the PF algorithm.

Table 1 presents RMSEs associated with the three filtering algorithms with Gaussian distributed noise. The RMSE of the PF is about 1.2 times smaller than that of both the UKF and the EKF for the small variance. The RMSE values of the EKF and the UKF, in case of the medium variance measurements, are 2 times and 5 times larger than that of the PF, respectively. Likewise, the RMSEs of the UKF and the EKF, in case of the high variance measurement, are four times and six times as high as that of the PF, respectively. This shows that the function gets more non-linear if the variances rise.

Table 1: The RMSE of estimated results in case of Gaussian noise.

STDs of uncertainties for distance and bearing	RMSE (m) Using Gaussian noise		
	PF	UKF	EKF
$\sigma_s = 0.1[m]$; $\sigma_\varphi = 0.02[rad]$	0.1059	0.1265	0.1260
$\sigma_s = 0.5[m]$; $\sigma_\varphi = 0.02[rad]$	0.1740	0.3792	0.8920
$\sigma_s = 1.0[m]$; $\sigma_\varphi = 0.02[rad]$	0.2123	0.7614	1.3679

Table 2: The RMSE of estimated results in case of non-Gaussian noise (triangular distributed noise).

STDs of uncertainties for distance and bearing	RMSE (m) Using triangular distributed noise		
	PF	UKF	EKF
$\sigma_s = 0.1[m]$; $\sigma_\varphi = 0.02[rad]$	0.1025	0.1500	0.1775
$\sigma_s = 0.5[m]$; $\sigma_\varphi = 0.02[rad]$	0.1802	0.6593	1.2325
$\sigma_s = 1.0[m]$; $\sigma_\varphi = 0.02[rad]$	0.2246	1.0502	2.0617

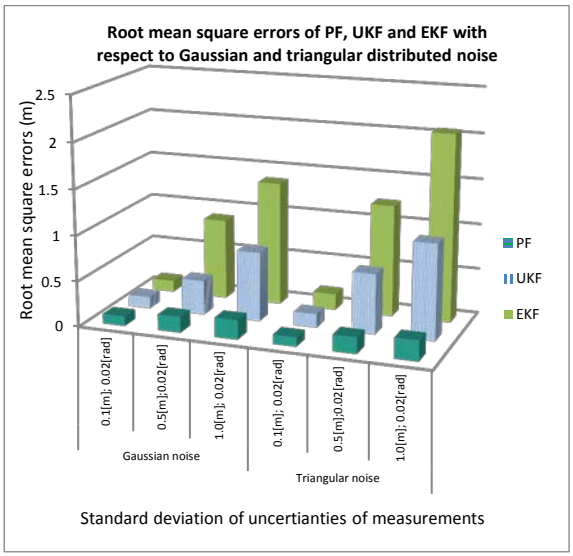

Figure 4: The RMSEs of estimated results in case of Gaussian noise compared to those in case of the triangular distributed noise.

Table 2 shows RMSEs for three algorithms by using triangular distributed noise. With respect to a small measurement variance, the RMSE value of the EKF is around 1.2 times and 1.7 times higher than that of the UKF and the PF, respectively. The RMSE of the PF is four times and seven times lower compared to those of the UKF and the EKF for the medium measurement variance. Similarly, in case of high measurement variance, the RMSE of the PF decreases five times and nine times as much as those of the UKF and the EKF, respectively.

Figure 4 shows a distinction of RMSEs for the three algorithms in case of Gaussian noise and the triangular distributed noise that data is organized in table 1 and table 2. It is clear that the RMSE of the PF depends only on the proportion of measurement variance but not on the probability distribution of measurement noise. In contrast, the RMSE of the EKF and the UKF has been strongly affected not only by the proportion of measurement variance, but also by the probability distribution of measurement noise.

4.2 Comparison of Computational Time

The computational complexity $O(\cdot)$ of each algorithm is considered as a combination of all computational processing steps. The EKF requires only one computational step of prediction and measurement function to obtain estimated values, while the PF demands *N* steps.

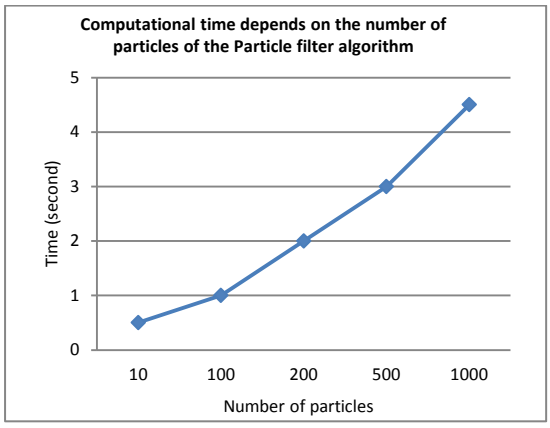

Figure 5: Computational time of PF varying by the number of particles using MATLAB R2014 running on Windows XP service Pack 3 with a 2.66GHz Intel Dual Core processor and 4GB RAM.

Therefore, the first order approximation of the EKF for large n_x (the number of total evaluation steps for both prediction and measurement function) is $O(n_x^3)$ and PF is typically $O(Nn_x^2)$, (Gustafsson, 2002). *N* is the number of particles. The UKF propagates more than one point through the non-linear prediction function ϕ and observation function φ. Therefore, the computational time of the UKF is about $O(n)$ higher than that of the EKF (Jose', 2013) where *n* denotes the dimension of the state vector. A comparison of the PF, the UKF and the EKF in terms of the computational time is realized for the non-linear prediction and measurement model with Gaussian noise for 100 measuring epochs. Figure 5 illustrates the computational time of the PF with respect to the change of the number of particles whereas the computational time of the EKF and the UKF is constant 7.5×10^{-4} seconds and 5×10^{-3} seconds, respectively.

5 CONCLUSIONS

In terms of accuracy, the PF works more accurately than the UKF and the EKF when the non-linear model is affected by Gaussian noise as well as non-Gaussian noise. While the PF is hardly influenced by the distribution of the measurement noise, the estimated results of the UKF and the EKF using Gaussian noise are about 1.5 times smaller than the results using non-Gaussian noise. Besides, the accuracy of filtering algorithms also depends on the proportion of the standard deviation of measurement noise. The PF obtains a significant improvement of the accuracy in comparison with the two remaining methods: If the measurement contains a larger standard deviation, the accuracy of the PF is about five times and nine times better than that of the UKF and EKF, respectively. The UKF also achieves an accuracy almost two times better than the EKF for the high measurement variance. But the accuracy of UKF and that of EKF is comparable in case of the small measurement variance. These results coincides with the theoretical base for the three approaches; thus delivering non-surprising results. The grade of non-linearity depends on the size of the variances for these non-linear functional relationships. For small variances, no linearity can be detected empirically.

In terms of computational time, the EKF is the fastest method followed by the UKF while the PF is more time consuming because of the number of particles.

In the future, the three algorithms will be investigated to identify the influence of the level of non-linearity as well as the probability distribution of the measurement noise on the estimated results. Furthermore, it would be valuable to reduce the computational time of the PF algorithm.

REFERENCES

Alkhatib, H., Neuman, I., Neuner, H., Kutterer, H., 2008. *Comparison of Sequential Monte Carlo Filtering with Kalman Filtering for Non-linear State Estimation.* 1st International Conference on Machine Control and Guidance.

Aussems, T., 1999. *Positionsschätzung von Landfahrzeugen mittels Kalman-Filterung aus Satelliten- und Koppelnavigationsbeobachtungen,* Veröffentlichung des Geodätischen Instituts der Rheinisch-Westfälischen Technischen Hochschule.

Douc, R., Olivier, C., Eric, M., 2005. *Comparison of resampling schemes for particle filtering.* Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis,.

Efron, B., 1979. Bootstrap methods: Another look at the Jackknife. *The annals of Statistics*, 7(1), pp. 1-26.

Gordon, N., Salmond, D., Smith, M., 1993. Novel approach to non-linear/non-Gaussian Bayesian state estimation. *IEE Proceedings*, 140(2), p. 107–113.

Gustafsson, F., 2002. Particle Filter Theory and Practice with Positioning Applications. *IEEE Transactions on Signal Processing*, 50(2), pp. 425-437.

Jose', L., Juan, A., 2013. *Simultaneous Localization and Mapping for Mobile Robots- introduction and method.* 1 edition. USA: IGI Global.

Julier, S., Uhlmann, K., 1997. *A New Extension of the Kalman Filter to Non-linear Systems.* Seattle, Washington, In The Proceedings of the American Control Conference.

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

Kalman, R., 1960. A new approach to Linear Filtering and Prediction Problems. *Transaction of the ASME-Journal of Basic Engineering,* pp. 33-45.

Liu, J., Chen, R., 1998. Sequential Monte Carlo Methods for Dynamic systems. *Journal of the American Statistical Association,* 93(443), pp. 1032-1044.

Pitt, M., Shephard, N., 1999. Filtering Via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association (American Statistical Association),* 94(446), pp. 590-591.

Ramm, K., 2008. *Evaluation von Filter-Ansätzen für die Positionsschätzung von Fahrzeugen mit den Werkzeugen der Sensitivitätsanalyse,* Reihe C, Nr. 619: DGK.

Särkkä, S., 2013. *Bayesian filtering and smoothing*. 1 edition. Cambridge University Press.

Schweitzer, J., 2012. *Modular Positioning using Different Motion Models.* Stuttgart, Germany, In the Proceedings on 3rd International Conference on Machine Control and Guidance.

Sternberg, H., 2000. *Zur Bestimmung der Trajektorie von Lanfahrzeugen mit einem hybriden Meßsystem,* Heft 67, Neubiberg: Schriftenreihe des Studiengangs Vermessungwesen der Universität der Bundeswehr München.

Wan, A., Van der Merwe, R., 2002. The Unscented Kalman Filter. Dans: *Kalman Filtering and Neural networks.* New York: Wiley, pp. 221-282.

Welch, G., Bishop, G., 1995. *An introduction to the Kalman filter,* Chapel Hill, USA: Technical report TR 95-041 University of North Carolina.