COMPARISON OF DIFFERENT APPROACHES TO ADAPTATION OF UNSCENTED KALMAN FILTER FOR NON-ODOMETRY SLAM

Jan Klečka

Doctoral Degree Programme (4), FEEC BUT

E-mail: xkleck01@stud.feec.vutbr.cz

Supervised by: Karel Horák

E-mail: horak@feec.vutbr.cz

Abstract: This paper is aimed at utilizing of an Unscented Kalman filter (UKF) for application on non-odometry SLAM problem. Non-odometry SLAM problem lacks state transitions function for part of state vector so theoretical part of the paper describes the derivation of three proposed modifications. The experimental part then shows experiments which evaluate modifications performance in reference to Extended Kalman filter. Experiments are based on simulation of monocular photogrammetry reconstruction problem.

Keywords: Unscented Kalman filter, UKF, SLAM, Recurrent estimation

1 INTRODUCTION

Simultaneous Localization and Mapping (SLAM) algorithms are an approximately three-decades-old scientific topic which has originally sprung from demands of mobile robotics. The basic concept is that observer i.e. robot moves and periodically makes observations of surrounding environment in order to get its position and virtual model of explored environment i.e. map. SLAM algorithms are then the methods used to the recurrent processing of environment observations into position and map. The methods recurrence is essential for algorithms real-time feasibility. [1]

Even though SLAM origins are strictly related to robots, and such platform almost directly imply that algorithm will have access to odometry data, nowadays no every device which utilizes such algorithm is capable to provide data for any motion model. SLAM algorithms capable of function without any motion model I refer as non-odometry. Non-odometry SLAM has compared to odometry SLAM its own challenges. The one I consider as crucial is the marginalization of the obsolete position information. It is a process which highly depends on the position estimate consistency and typically the algorithms based on the propagation of probability using linearization fails here.

In this paper, I describe my process of utilizing of a fairly new modification of Kalman filter referred as Unscented for non-odometry. This modification realizes the uncertainty propagation on deterministic sampling and I want to explore its performance because when compared in the context of other non-linear modifications of Kalman filter is often referred as ‘slightly better’.

2 MATHEMATICAL BACKGROUND

Let’s briefly approach the online non-odometry SLAM problem from the probabilistic point of view. The recurrent update step is defined as:

$$p(\mathbf{x}_i, \mathbf{m} \mid \mathbf{z}_{0:i}) = \eta p(\mathbf{z}_i \mid \mathbf{x}_i, \mathbf{m}) \int p(\mathbf{x}_{i-1}, \mathbf{m} \mid \mathbf{z}_{0:N-1}) d\mathbf{x}_{i-1} \quad (1)$$

Where $\mathbf{x}_i$ is vector describing position and orientation in time i, $\mathbf{m}$ is a mathematical representation of the environment and $\mathbf{z}_{0:i}$ is set of observations captured from time to time i. Convenient for noticing is the integral term of the equation which represent the marginalization step.

$$p(\mathbf{m} \mid \mathbf{z}_{0:i-1}) = \int p(\mathbf{x}_{i-1}, \mathbf{m} \mid \mathbf{z}_{0:i-1}) d\mathbf{x}_{N-1} \quad (2)$$

However, result in form of the probability distribution is challenging either to obtain and also to work with. The Kalman filter simplifies the problem by estimating two statistical moments of the estimate's distribution – the mean and the covariance.

2.1 STANDARD UKF

For comparison and because of several definitions I firstly present the standard UKF algorithm which is in detail described in [2]. Let's have a non-linear dynamic system:

$$\boldsymbol{\theta}_i = \mathrm{f}(\boldsymbol{\theta}_{i-1}, \mathbf{u}_i) + w \quad (3)$$

$$\mathbf{z}_i = \mathrm{h}(\boldsymbol{\theta}_i) + v \quad (4)$$

Where v and w are stochastic variables with zero mean and covariance $E[vv^T] = \mathbf{R}$ $E[ww^T] = \mathbf{Q}$.

Then the first part is a prediction step:

$$\hat{\boldsymbol{\theta}}_{i|i-1} = \sum_{k=0}^{2L} W_k^{(m)} \mathrm{f}(\boldsymbol{\chi}^{\boldsymbol{\theta}}_{i-1|i-1,k}, \mathbf{u}_i) \quad (5)$$

$$\mathbf{P}_{i|i-1} = \mathbf{Q} + \sum_{k=0}^{2L} W_k^{(c)} \left(\mathrm{f}(\boldsymbol{\chi}^{\boldsymbol{\theta}}_{i-1|i-1,k}, \mathbf{u}_i) - \hat{\boldsymbol{\theta}}_{i|i-1}\right)\left(\mathrm{f}(\boldsymbol{\chi}^{\boldsymbol{\theta}}_{i,N-1|N-1}, \mathbf{u}_N) - \hat{\boldsymbol{\theta}}_{i|i-1}\right)^T \quad (6)$$

Where $L = \dim(\boldsymbol{\theta})$, $\lambda = \alpha^2(L+\kappa)$, α, β, κ are some free parameters, $W_k^{(m,c)}$ is weights defined as: $W_0^{(m)} = \frac{\lambda}{L+\lambda}$, $W_0^{(c)} = \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta$, $W_k^{(c,m)} = \frac{1}{L+\lambda} \forall k \neq 0$. And the variable $\boldsymbol{\chi}$ is set of so-called sigma points which are mathematically defined:

$$\boldsymbol{\chi}^{\theta}_{i-1|i-1,0:2L} = \left(\hat{\boldsymbol{\theta}}_{i-1|i-1} \mid \hat{\boldsymbol{\theta}}_{i-1|i-1} \pm \sqrt{(L+\lambda)\mathbf{P}_{i-1|i-1}}\right) \quad (7)$$

Where the term $\sqrt{(L+\lambda)\mathbf{P}_{i-1|i-1}}$ is the square root of a matrix. By [3] is recommended to compute it using Cholesky decomposition for its numerical stability.

This step is followed by update step which begins with determining the optimum value of Kalman gain

$$\hat{\mathbf{z}}_i = \sum_{k=0}^{2L} W_k^{(m)} \mathrm{h}(\boldsymbol{\chi}^{\boldsymbol{\theta}}_{i|i-1,k}) \quad (8)$$

$$\mathbf{S}_{i|i-1} = \mathbf{R} + \sum_{k=0}^{2L} W_k^{(c)} \left(\mathbf{h}(\boldsymbol{\chi}^{\boldsymbol{\theta}}_{i|i-1,k}) - \hat{\mathbf{z}}_i\right)\left(\mathbf{h}(\boldsymbol{\chi}^{\boldsymbol{\theta}}_{i|i-1,k}) - \hat{\mathbf{z}}_i\right)^T \quad (9)$$

$$\mathbf{K}_i = \left[\sum_{k=0}^{2L} W_k^{(c)} \left(\boldsymbol{\chi}^{\boldsymbol{\theta}}_{i|i-1,k} - \hat{\boldsymbol{\theta}}_{i|i-1}\right)\left(\mathbf{h}(\boldsymbol{\chi}^{\boldsymbol{\theta}}_{i|i-1,k}) - \hat{\mathbf{z}}_i\right)^T\right]\mathbf{S}^{-1}_{i|i-1} \quad (10)$$

After obtaining of Kalman gain is the algorithm same as the original Kalman filter:

$$\hat{\boldsymbol{\theta}}_{i|i} = \hat{\boldsymbol{\theta}}_{i|i-1} + \mathbf{K}_i(\mathbf{z}_i - \hat{\mathbf{z}}_i) \quad (11)$$

$$\mathbf{P}_{i|i} = \mathbf{P}_{i|i-1} - \mathbf{K}_i \mathbf{S}_{i|i-1} \mathbf{K}_i^T \quad (12)$$

2.2 UKF FOR NON-ODOMETRY SLAM

The first main difference from standard application came from lack of motion model. Because there is no link between two consequent state vectors $p(\mathbf{x}_i \mid \mathbf{x}_{i-1}) = p(\mathbf{x}_i)$ the prediction step contains only the marginalization which the estimate is represented by a mean vector and a covariance matrix is a simple selection of a subvector and a submatrix corresponding strictly to map parametrization.

$$\hat{\mathbf{m}}_{i-1} = (\mathbf{0} \mid \mathbf{I}) \begin{pmatrix} \hat{\mathbf{x}}_{i-1} \\ \hat{\mathbf{m}}_{i-1} \end{pmatrix} \tag{13}$$

$$\mathbf{P}_{i|i-1} = (\mathbf{0} \mid \mathbf{I}) \mathbf{P}_{i-1|i-1} (\mathbf{0} \mid \mathbf{I})^T \tag{14}$$

Then the update step:

$$\hat{\mathbf{z}}_i = \sum_{k=0}^{2L} W_i^{(m)} \, \mathrm{h}\left(\boldsymbol{\chi}^x_{i|i-1,k}, \boldsymbol{\chi}^m_{i|i-1,k}\right) \tag{15}$$

$$\mathbf{S}_{i|i-1} = \mathbf{R} + \sum_{k=0}^{2L} W_i^{(c)} \left(\mathrm{h}\left(\boldsymbol{\chi}^x_{i|i-1,k}, \boldsymbol{\chi}^m_{i|i-1,k}\right) - \hat{\mathbf{z}}_i\right)\left(\mathrm{h}\left(\boldsymbol{\chi}^x_{i|i-1,k}, \boldsymbol{\chi}^m_{i|i-1,k}\right) - \hat{\mathbf{z}}_i\right)^T \tag{16}$$

$$\mathbf{K}_i = \left[\sum_{i=0}^{2L} W_i^{(c)} \left(\begin{pmatrix} \boldsymbol{\chi}^x_{i|i-1,k} \\ \boldsymbol{\chi}^m_{i|i-1,k} \end{pmatrix} - \begin{pmatrix} \sum_{k=0}^{2L} W_i^{(m)} \boldsymbol{\chi}^x_{i|i-1,k} \\ \hat{\mathbf{m}}_{i-1} \end{pmatrix} \right) \left(\mathrm{h}\left(\boldsymbol{\chi}^x_{i|i-1,k}, \boldsymbol{\chi}^m_{i|i-1,k}\right) - \hat{\mathbf{z}}_i\right)^T \right] \mathbf{S}^{-1}_{i|i-1} \tag{17}$$

$$\begin{pmatrix} \hat{\mathbf{x}}_{i|i} \\ \hat{\mathbf{m}}_i \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{x}}_{i|i-1} \\ \hat{\mathbf{m}}_{i-1} \end{pmatrix} + \mathbf{K}_i \left(\mathbf{z}_i - \hat{\mathbf{z}}_i\right) \tag{18}$$

$$\mathbf{P}_{i|i} = \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{P}_{i|i-1} \end{pmatrix} - \mathbf{K}_i \mathbf{S}_{i|i-1} \mathbf{K}_i^T \tag{19}$$

The last thing left to define is a way to compute sigma points. Obtainment of sigma points for $\mathbf{m}$ is fairly straightforward – as in standard algorithm the square root of covariance matrix can be used:

$$\boldsymbol{\chi}^m_{i|i-1,0:2L} = \left(\hat{\mathbf{m}}_{i-1} \mid \hat{\mathbf{m}}_{i-1} \pm \sqrt{(L+\lambda)\mathbf{P}_{i|i-1}}\right) \tag{20}$$

However, about the variable $\mathbf{x}_i$, we have initially no information so there is no mean and covariance matrix to utilize. This is the core problem of modifying UKF for non-odometry SLAM and so main subject of following experiments. I developed three ways to compute these sigma points.

Firstly, the mean is computed for all variants using the Maximum likelihood method:

$$\hat{\mathbf{x}}_{i|i-1} = \operatorname*{argmax}_{\mathbf{x}_i} \left(\ell\left(\mathbf{x}_i \mid \mathbf{z}_i, \hat{\mathbf{m}}_{i-1}\right)\right) \tag{21}$$

Constant value:

As the name probably suggest the first method doesn't take into consideration any variability of position while varying the map representation:

$$\boldsymbol{\chi}^x_{i|i-1,k} = \hat{\mathbf{x}}_{i|i-1} \tag{22}$$

Linear correction:

The second method uses a linear transformation to reflect changes in the map into position estimation. The intuition for it is based on first-order Taylor expansion around $\hat{\mathbf{x}}_{i|i-1}$ of observation function in the derivative of (21) based on normally distributed noise assumption

$$\frac{\partial \ell\left(\mathbf{x}_i \mid \mathbf{z}_i, \hat{\mathbf{m}}_{i-1}\right)}{\partial \mathbf{x}_i} = \mathbf{H}_x^T \mathbf{R}^{-1} \left(\mathrm{h}\left(\hat{\mathbf{x}}_{i|i-1}, \hat{\mathbf{m}}_{i-1}\right) + \mathbf{H}_x \left(\chi^x_{i|i-1,k} - \hat{\mathbf{x}}_{i|i-1}\right) + \mathbf{H}_m \left(\chi^m_{i|i-1,k} - \hat{\mathbf{m}}_{i-1}\right) - z\right) \tag{23}$$

Which for ML estimate ($\boldsymbol{\chi}^{x}_{i|i-1,k} = \hat{\mathbf{x}}_{i|i-1}$ and $\boldsymbol{\chi}^{m}_{i|i-1,k} = \hat{\mathbf{m}}_{i-1}$) is equal to zero vector. To be zero-valued also for non-mean values of $\boldsymbol{\chi}^{m}_{i|i-1,k}$ we can compensate by substituting

$$\boldsymbol{\chi}^{x}_{i|i-1,k} = \hat{\mathbf{x}}_{i|i-1} - \left(\mathbf{H}_x^T \mathbf{R}^{-1} \mathbf{H}_x\right)^{-1} \mathbf{H}_x^T \mathbf{R}^{-1} \mathbf{H}_m \left(\boldsymbol{\chi}^{m}_{i|i-1,k} - \hat{\mathbf{m}}_{i-1}\right) \tag{24}$$

Maximum likelihood:

The third and the last method I proposed to compute sigma points for position variable utilizes the maximum likelihood method:

$$\boldsymbol{\chi}^{x}_{i|i-1,k} = \underset{\mathbf{x}_i}{\operatorname{argmax}} \left(\boldsymbol{\ell}\left(\mathbf{x}_i \mid \mathbf{z}_i, \boldsymbol{\chi}^{m}_{i,N|N-1}\right)\right) \tag{25}$$

3 EXPERIMENTS

To evaluation performance of different variants of position sigma points generation, I decided to realize a simulation based on the reconstruction of a virtual environment using the monocular camera model. I did so because monocular reconstruction is a highly non-linear process and so I assumed that any differences in performance will be clearly observable.

I created a virtual environment composed of 32 3D points. These points were uniformly spaced and lied on two planes. The camera trajectory was a spiral around environment points and the camera is always oriented in a way that its optical axis perpendicularly intersects the horizontal axis of environment frame and I generated a set of 200 observations with positions and orientations are evenly spaced on this trajectory. This set of perfect observations was then degraded by adding a 100 different realizations of normally distributed noise to it. And this was my testing data: 100 sets of 200 noised observations.

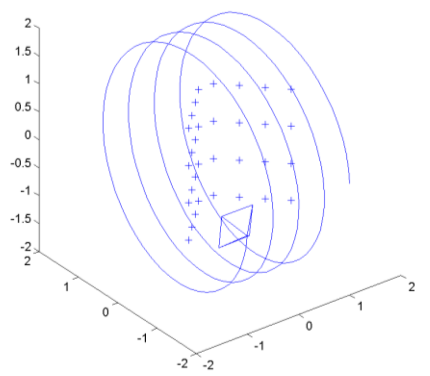

Figure 1: Virtual environment and camera trajectory

However, every recurrent algorithm needs an initial estimate. To expand the scope of my experiments I used 5 initial estimates to evaluate the performance of proposed variants. First, two initial estimates were given by adding a Gaussian noise to environment points one with $\sigma = 0.01$ and the second with $\sigma = 0.1$. Last three initial estimates were results of maximum likelihood method which uses the first n observations. The n was 4,16 and 64 respectively. Using this five ways of generating initial estimates I processed all 100 set of observations and compute two statistics. First was the sum of estimations errors for each point in the Euclidian norm, which should express how precise is the result.

$$Err_{Euclid} = \sum_{k=1}^{32} \left\| \mathbf{m}_k - \hat{\mathbf{m}}_{200,k} \right\| \tag{26}$$

Where $\mathbf{m}_k$ is the k-th point from the set of environment points.

And the second statistic should represent whether is the estimate consistent with reality. I commuted the Mahalanobis distance and normalize it by dimensionality which assuming the normally distributed reconstruction error should be unit in average:

$$Err_{Mahanobis} = (\hat{\mathbf{m}}_{200} - \mathbf{m})^T \mathbf{P}_{200}^{-1} (\hat{\mathbf{m}}_{200} - \mathbf{m}) / \dim(\mathbf{m}) \tag{27}$$

The results are of following graphs. Because large differences I had to put then into logarithmical scale.

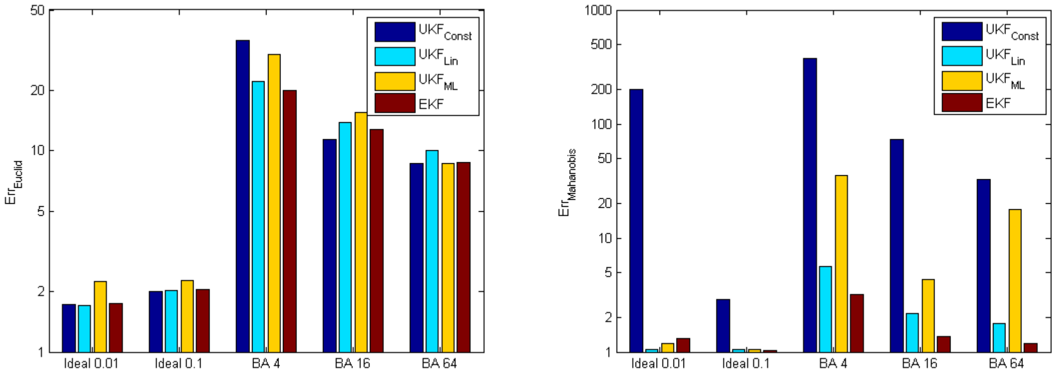

Figure 2: Median of Euclid (left) and Mahalanobis (right) errors

4 CONCLUSION

During our experiments, the UKF algorithm did not show any properties which would make it superior to EKF. The first variant produces estimates highly inconsistent with reality because no variability in position variable causes loss of uncertainty from position estimation. The third method is compared to the others computational significantly more demanding and also fairly unstable – sometimes nonlinear optimization during sigma points computation fails and occasionally this variant produces covariance matrix which is not positive-definite.

ACKNOWLEDGEMENT

The completion of this paper was made possible by the grant No. FEKT-S-17-4234 - „Industry 4.0 in automation and cybernetics" financially supported by the Internal science fund of Brno University of Technology and Competence Center realized by TACR (reg. number TE01020197).

REFERENCES

[1] DURRANT-WHYTE, H. and T. BAILEY. Simultaneous localization and mapping (SLAM): Part I. IEEE Robotics and Automation Magazine [online]. 2006, 13(2), 99-108

[2] WAN, E.A. and R. VAN DER MERWE. The unscented Kalman filter for nonlinear estimation. In: Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000 [online]. IEEE Publishing, 2000, s. 153-158 ISBN 0-7803-5800-7.

[3] JULIER, S.J. and J.K. UHLMANN. Unscented filtering and nonlinear estimation. Proceedings of the IEEE [online]. USA: IEEE, 2004, 92(3), 401-422 [cit. 2018-03-09]. DOI: 10.1109/JPROC.2003.823141. ISSN 0018-9219.