Bulletin of the *Transilvania* University of Braşov • Vol. 3 (52) - 2010
Series I: Engineering Sciences

UNSCENTED KALMAN FILTER POSITION ESTIMATION FOR AN AUTONOMOUS MOBILE ROBOT

C. SULIMAN[1] **F. MOLDOVEANU**[1]

Abstract: *The Kalman filters have been widely used for mobile robot navigation and system integration. So that it may operate autonomously, a mobile robot must know where it is. Accurate localization is a key prerequisite for successful navigation in large-scale environments, particularly when global models are used, such as maps, drawings, topological descriptions, and CAD models. This paper presents the localization of a mobile robot using one variation of the traditional Kalman filter: the unscented Kalman filter (UKF). For this purpose the filter was implemented for a known kinematic model of the robot.*

Key words: *autonomous mobile robot, Kalman filter, unscented Kalman filter, position estimation.*

1. Introduction

Filtering is a very used method in engineering and embedded systems. A good filtering algorithm can reduce the noise from signals while retaining the useful information. The Kalman filter (KF) [8] is a mathematical tool that can estimate the variables of a wide range of process. It estimates the states of a linear system. This type of filter works very well in practice and that is why it is often implemented in embedded control system and because we need an accurate estimate of the process variables. The KF has been widely used for mobile robot navigation [1], [2] and system integration. So that it may operate autonomously, a mobile robot must know where it is. Accurate localization is a key perquisite for successful navigation in large-scale environments, particularly where global models are used. The KF has many limitations and that is why many authors proposed various fixes and modifications to better estimate the process variables [4].

One of the many variations of the Kalman filter is the unscented Kalman filter (UKF). This filter is based on the unscented transform (UT) [5]. The UKF works by approximating a Gaussian distribution, which is much easier than approximating an arbitrary nonlinear function. The UKF is computationally more costly than the EKF, but it reduces estimation errors. One of the advantages of the UKF over the EKF is that it doesn't need to derive the Jacobian matrices. The most important application of the UKF is

[1] Dept. of Automatics, *Transilvania* University of Braşov.

in simultaneous localization and map building (SLAM). It has been applied to ground mobile robots [3], [6], but has been successfully applied even to unmanned aerial vehicles (UAV) [7].

This paper presents the implementation for the unscented Kalman filter for an autonomous mobile robot based on Ackerman steering. The simulations in this paper will show the accuracy of the UKF. In section 2 of the paper we will present the traditional KF. In section 3 we present the of the UKF implementation for an autonomous mobile robot. The last two sections of the paper contain the simulation results and the conclusions drawn from these simulations and future work.

2. The Kalman Filter

The Kalman filter is composed from a set of mathematical equations that provide the computational means for estimating the state of a process, in a way that minimizes the mean of the squared error. This type of filter is very powerful because: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown.

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of noisy measurements. The equations of the Kalman filter fall into two categories: time update equations and measurement update equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimates for the next time step. The measurement update equations are responsible for the feedback, for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate.

The specific equations for the prediction step (time and measurement updates) are:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k , \tag{1}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k , \tag{2}$$

where: F_k is the state transition model which is applied to the previous state $\hat{x}_{k-1|k-1}$; B_k is the control-input model which is applied to the control vector u_k. From the above equations we can see how the state and covariance estimates are projected forward in time, from the time step $k - 1$ to step k.

In the observation step we have:

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} , \tag{3}$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k , \tag{4}$$

where: $\tilde{y}_k$ is called the innovation or measurement residual; H_k is the observation model which maps the true state space into the observed space; $P_{k|k-1}$ represents the predicted (a priori) estimate covariance; Q_k is the covariance of the process noise.

For the measurement update step we have:

$$K_k = P_{k|k-1} H_k^T S_k^{-1} , \tag{5}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k , \tag{6}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} . \tag{7}$$

where: K_k is the optimal Kalman gain and S_k is the innovation (or residual) covariance. The first task during the measurement update step is to compute the Kalman gain (5). The next step is to actually measure the process to obtain z_k, and then to generate an a posteriori state estimate by incorporating the measurement as in (6). The final step is to obtain an a posteriori error covariance estimate via (7).

After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to project or predict the new a priori estimates.

3. The UKF Implementation

To model the robot position, we wish to know its *x* and *y* coordinates and its orientation. These three parameters can be combined into a vector called a state variable vector. The robot uses an overhead camera to obtain the information about how far the robot has traveled into the environment so that it can calculate its position. These measurements include a component of error. If trigonometry is used to calculate the robot's position it can have a large error and can change significantly from frame to frame depending on the measurement at the time.

For the UKF implementation we have used the kinematic model of an autonomous mobile robot based on Ackermann steering (see Figure 1). The mobile robot is supposed to move in a 2D coordinate system.

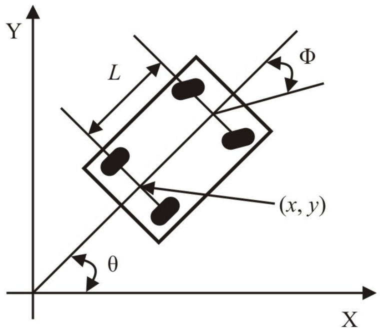

Fig. 1. *The autonomous mobile robot representation in a 2D coordinate system*

The kinematic equations for the mobile robot based on Ackermann steering are presented in the following:

$$\dot{x} = V \cdot \cos\theta\,, \tag{8}$$

$$\dot{y} = V \cdot \sin\theta\,, \tag{9}$$

$$\dot{\theta} = \frac{V \cdot \tan\Phi}{L}\,. \tag{10}$$

In the above system of equations *V* is the velocity of the robot, *L* represents the distance between the rear axel and front one, and Φ is the steering angle.

From the nonlinear system of equations presented above, we deduced that we cannot use the traditional form of the KF, instead we use the UKF form.

The problem of propagating a Gaussian random variable through a nonlinear function can also be approached using the unscented transform. Instead of linearizing the function, this method uses a set of points and propagates the through the nonlinear function, and thus eliminating the process of linearization (see Figure 2). The points are chosen in such manner that their mean, covariance and possibly their higher order moments match the Gaussian random variable.

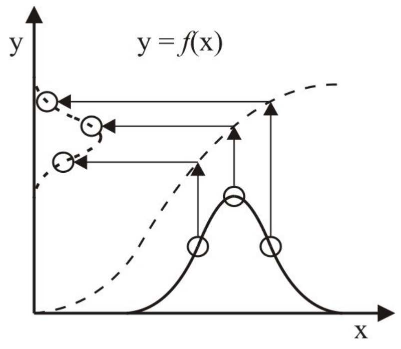

Fig. 2. *Unscented Kalman filter propagation of sigma points from a Gaussian distribution through a nonlinear function, and recreation of the Gaussian distribution by computing the mean and covariance of the results*

The mean and the covariance can be recalculated from the propagated points, and thus we can obtain more accurate result compared to the ordinary function linearization. The main idea is also to approximate the probability distribution instead of the function. This strategy decreases the computational complexity and, at the same time, it increases estimate accuracy, thus giving more accurate results.

The filter starts by augmenting the state vector to L dimensions, where L is the sum of dimensions in the original state-vector, model noise and measurement noise. The covariance matrix is similarly augmented to a L^2 matrix. Together this forms the augmented state estimate vector $\hat{x}^a$ and covariance matrix $\hat{P}^a$:

$$\hat{x}^a_{k-1} = \begin{bmatrix} \hat{x}^T_{k-1} \\ 0 \\ 0 \end{bmatrix}, \tag{11}$$

$$P^a_{k-1} = \begin{bmatrix} P_{k-1} & 0 & 0 \\ 0 & Q_{k-1} & 0 \\ 0 & 0 & R_{k-1} \end{bmatrix}. \tag{12}$$

The next step consists of creating $2L + 1$ sigma-points in such a way that they together capture the full mean and covariance of the augmented state. The X matrix is chosen to contain these points, and its columns are calculated as follows:

$$\bar{x} = x^a_{k-1|k-1}, \quad X^0_{k-1|k-1} = \bar{x}, \tag{13}$$

$$X^i_{k-1|k-1} = \bar{x} + \left(\sqrt{(L+\lambda)P^a_{k-1|k-1}}\right)_i, \quad i = 1, ..., L, \tag{14}$$

$$X^i_{k-1|k-1} = \bar{x} - \left(\sqrt{(L+\lambda)P^a_{k-1|k-1}}\right)_{i-L}, \quad i = L+1, ..., 2L, \tag{15}$$

where: $\bar{x}$ and $P^a_{k-1|k-1}$ represent the mean and covariance of the random variable x.

Then the sigma points are propagated through the nonlinear function:

$$X^i_{k|k-1} = f(X^i_{k-1|k-1}), \; i = 0, ..., 2L, \tag{16}$$

from which the mean and covariance can be approximated:

$$\hat{x}_{k|k-1} = \sum_{i=0}^{2L} W^i_m X^i_{k|k-1}, \tag{17}$$

$$P_{k|k-1} = \sum_{i=0}^{2L} W^i_c (X^i_{k|k-1} - \hat{x}_{k|k-1})(X^i_{k|k-1} - \hat{x}_{k|k-1})^T \tag{18}$$

Each sigma-point is also assigned a weight. These weight are derived by comparing the moments of the sigma-points with a Taylor series expansion of the models while assuming a Gaussian distribution.

The resulting weights for mean m and covariance c estimates then becomes:

$$W^0_m = \frac{\lambda}{L+\lambda}, \tag{19}$$

$$W^0_c = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta), \tag{20}$$

$$W^i_m = W^i_c = 0.5(L+\lambda), \; i = 1, ..., 2L, \tag{21}$$

$$\lambda = \alpha^2 (L+k) - L. \tag{22}$$

In the above equations α and k control the spread of the sigma points and β is related to the distribution of x. The α parameter is chosen in the interval $0 < \alpha \le 1$, and it is set to a low value, 0.001, to avoid non-local effects.

When we first use the filter we need to initialize it:

$$\hat{x}_0^a = \begin{bmatrix} \hat{x}_0^T \\ 0 \\ 0 \end{bmatrix} \text{ and } P_0^a = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix}.$$

The filter then predicts next state by propagating the sigma-points through the state and measurement models, and then calculating weighted averages and covariance matrices of the results:

$$\gamma_{k|k-1}^i = h(X_{k|k-1}^i),\ i = 0...2L, \tag{23}$$

$$\hat{z}_{k|k-1} = \sum_{i=0}^{2L} W_m^i \gamma_{k|k-1}^i . \tag{24}$$

The predictions are then updated with new measurements by first calculating the measurement covariance and state-measurement cross correlation matrices, which are then used to determine the Kalman gain:

$$P_{zz} = \sum_{i=0}^{2L} W_c^i \left[\gamma_{k|k-1}^i - \hat{z}_{k|k-1}\right] \cdot \left[\gamma_{k|k-1}^i - \hat{z}_{k|k-1}\right]^T , \tag{25}$$

$$P_{xz} = \sum_{i=0}^{2L} W_c^i \left[X_{k|k-1}^i - \hat{x}_{k|k-1}\right] \cdot \left[\gamma_{k|k-1}^i - \hat{z}_{k|k-1}\right]^T , \tag{26}$$

$$K_k = P_{xz} P_{zz}^{-1} , \tag{27}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - \hat{z}_{k|k-1}) , \tag{28}$$

$$P_{k|k} = P_{k|k-1} - K_k P_{zz} K_k^T . \tag{29}$$

4. Experimental Results

In this section of the paper we show the results obtained by simulating UKF case for an autonomous mobile robot. For this purpose the UKF filter was implemented in the programming and simulation environment called Matlab. For our tests we have considered a very simple case: a robot that follows a predefined path. The position of the robot in the environment is obtained by using an overhead camera. On the robot we've put a square marker. To obtain the position of the marker we've developed an application in Visual C++, using the augmented reality toolbox ARToolkit. The data obtained from the marker detection application is fed into the UKF which will estimate the robot's position from the real noisy measurements. In the below figure it is presented the estimated path of the robot compared to the real path. The green line is the predefined path that was chosen randomly, and the red line is the estimated path. Also, in the simulations, the blue triangle represents the robot's real position and the red one is the robot's estimated position.

5. Conclusions

In this paper one of the main variations of the Kalman filter used for the position estimation of an autonomous mobile robot based on Ackermann steering. From the simulation results shown earlier we have deduced that the UKF has shown promising results. The difference in nonlinearity between systems encountered in everyday life can give a big difference in the results obtain from the presented filtering techniques. The advantage of the UKF over other variations of the classical KF increases with the degree of nonlinearity in the measurement model.

In the future we will try to combine the fuzzy logic the filter presented in this paper for a better position estimation. Furthermore, we will use the UKF in a visual based SLAM system, where camera calibration and correct feature detection play a vital role in solving the data association problem.

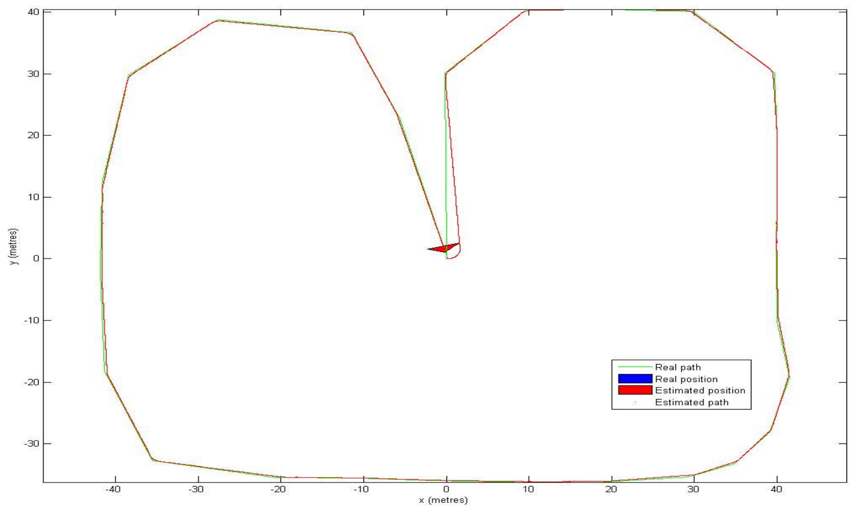

Fig. 3. *Trajectory estimation with the UKF*

Acknowledgement

This paper is supported by the Sectoral Operational Programme Human Resources Development (SOP HRD), financed from the European Social Fund and by the Romanian Government under the contract number POSDRU/6/1.5/S/6.

References

1. Choi, B.S., Lee, J.J.: *The Position Estimation of Mobile Robot under Dynamic Environment*. In: Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society, 2007, p. 134-138.
2. Hu, C., Chen, W., Chen, Y., Liu, D.: *Adaptive Kalman Filtering for Vehicle Navigation*. In: Journal of Global Positioning Systems **2** (2003) No. 1, p. 42-7.
3. Holmes, S., Klein, G., Murray, D.W.: *A Square Root Unscented Kalman Filter for Visual MonoSLAM*. In: Proceedings of IEEE International Conference on Robotics and Automation - ICRA, 2008, p. 3710-3716.
4. Johnson, R., Sasiadek, J., Zalewski, J.: *Kalman Filter Enhancement for UAV Navigation*. In: Proceedings of the International Symposium on Collaborative Technologies and Systems, 2003.
5. Julier, S., Ulhmann, J.K.: *A New Extension of the Kalman Filter to Nonlinear Systems*. In: Proceedings of Areosense: 11th International Symposium Aerospace/Defense Sensing, Simulation, and Controls **3068** (1997), p. 182-193.
6. Ko, S., Choi, J., Kim, B.: *Indoor Mobile Localization System and Stabilization of Localization Performance using Pre-filtering*. In: International Journal of Control, Automation, and Systems **6** (2008), p. 204-213.
7. Sünderhauf, N., Lange, S., Protzel P.: *Using the Unscented Kalman Filter in Mono-SLAM with Inverse Depth Parametrization for Autonomous Airship Control*. In: IEEE International Workshop on Safety, Security and Rescue Robotics, 2007, p. 1-6.
8. Welch, G., Bishop, G.: *An Introduction to the Kalman Filter*. In: Technical Report: TR95-041, University of North Carolina, 2006.