Kalman Filter Theory Based Mobile Robot Pose Tracking Using Occupancy Grid Maps

Edouard Ivanjko, Mario Vašak, and Ivan Petrović

Abstract—In order to perform useful tasks the mobile robot’s current pose must be accurately known. Problem of finding and tracking the mobile robot’s pose is called localization, and can be global or local. In this paper we address local localization or mobile robot pose tracking with prerequisites of known starting pose, robot kinematic and world model. Pose tracking is mostly based on odometry, which has the problem of accumulating errors in an unbounded fashion. To overcome this problem sensor fusion is commonly used. This paper describes two methods for calibrated odometry and sonar sensor fusion based on Kalman filter theory and occupancy grid maps as used world model. Namely, we compare the pose tracking or pose estimation performances of both most commonly used nonlinear-model based estimators: extended and unscented Kalman filter. Since occupancy grid maps are used, only sonar range measurement uncertainty has to be considered, unlike feature based maps where an additional uncertainty regarding the feature/range reading assignment must be considered. Thus the numerical complexity is reduced. Experimental results on the Pioneer 2DX mobile robot show similar and improved accuracy for both pose estimation techniques compared to simple odometry.

I. INTRODUCTION

Mobile robot localization is one of the very important tasks in navigation of autonomous mobile robots [1]. In a typical indoor environment with a flat floor plan, mobile robot localization becomes a matter of estimating its pose, its position given with its x and y coordinates and its orientation given with angle $\Theta$. There are two types of localization: global and local. Using global localization methods mobile robot can solve the unknown start pose and lost-robot problem, i.e. current pose in an a priori known world model can always be determined. On the other hand, using local localization or pose tracking the mobile robot becomes lost when predicted sensor readings become significantly different from real sensor readings. In spite of this drawback of pose tracking methods they are extensively used due to their computational simplicity, which is of particular importance in applications where all algorithms should be computed in onboard computer. A global localization algorithm should also be implemented, but its execution is started only in cases when robot is lost. In such a situation robot interrupts its normal operation and starts wandering through the environment trying to find its pose.

When it succeeds, execution of global localization algorithm is stopped and robot continues with normal operation.

One of the most important means of solving the pose tracking problem is odometry. This method uses encoder data and is a simple, inexpensive and easy way to determine the offset from a known start pose in real time. The encoder data are proceeded to the central processor that in turns continually updates the mobile robot’s pose using geometric equations. The disadvantage is its unbounded accumulation of errors due to wheel slippage, floor roughness, discretized sampling of wheel speed data, inaccessibility to the angular velocities of the wheels in some mobile robots etc. So this method can be successfully used only for pose tracking between pose updates using additional sensors [2].

A lot of research has been undergone in order to improve the accuracy of odometry, i.e. to eliminate the systematic error, mobile robot construction constraints and environment influences on the mobile robot pose tracking. A common approach consists of two parts. The first part is about a better error [3] or odometry model [4] and the second part involves usage of additional sensors [2]. Mostly used additional sensors are sonar, laser range finder, stereo or mono vision, gyro, compass and GPS (for outdoor mobile robots). Additional sensors can be used for estimated robot pose correction, for online odometry calibration [5], or for both simultaneously. In case of mobile robot pose correction local and global world model matching [6] or sensor fusion techniques are used [7], [8].

Used sensor fusion approach is based on nonlinear model based estimators: extended and unscented Kalman filter (EKF and UKF). The idea is to match recent sensory information against prior knowledge of the environment, i.e. world model which is in our case an occupancy grid map. Occupancy grid maps are the dominant paradigm for environment modelling in mobile robotics and they facilitate various
key aspects of mobile robot navigation, such as localization, path planning and collision avoidance. Due to problems with bad sonar placement on our mobile robot, a sonar based local map with high quality cannot be determined [9]. A local occupancy grid map obtained using our mobile robot is given in Fig 1. It can be noticed that only sonars perpendicular to surrounding walls give a consistent range reading. Other sonar readings are false and greatly reduce the map quality. Therefore a technique similar as in [7] is adopted that can find direct correspondences between mobile robot sensor readings and readings predicted from a world model. False sonar range readings are rejected by using a threshold comparison between predicted and measured sonar model. False sonar range readings are rejected by using a robot sensor readings and readings predicted from a world adopted that can find direct correspondences between mobile robot is given in Fig 1. It can be noticed that only sonars perpendicular to surrounding walls give a consistent range reading. Other sonar readings are false and greatly reduce the map quality. Therefore a technique similar as in [7] is adopted that can find direct correspondences between mobile robot sensor readings and readings predicted from a world map. False sonar range readings are rejected by using a threshold comparison between predicted and measured sonar model.

The paper is organized as follows. Section II presents used mobile robot kinematic model and applied calibration extension. Section III describes EKF and UKF based mobile robot localizations. Experimental results obtained using mentioned pose estimation techniques are presented and analyzed in Section IV.

II. MOBILE ROBOT MODEL

In the experiments we use a three-wheeled mobile robot. Two front wheels are drive wheels with encoder mounted on them, and the third wheel is a castor wheel needed for stability. Drive wheels can be controlled independently from each other. The kinematic model of the mobile robot is given by the following relations (Fig. 2):

\[
\begin{align*}
    x_{k+1} &= x_k + D_k \cdot \cos \Theta_{k+1}, \\
    y_{k+1} &= y_k + D_k \cdot \sin \Theta_{k+1}, \\
    \Theta_{k+1} &= \Theta_k + \Delta \Theta_k, \\
    D_k &= v_{t,k} \cdot T, \\
    \Delta \Theta_k &= \omega_k \cdot T, \\
    v_{l,k} &= \frac{v_{L,k} + v_{R,k}}{2} = \frac{\omega_{L,k} R + \omega_{R,k} R}{2}, \\
    \omega_k &= \frac{v_{R,k} - v_{L,k}}{b} = \frac{\omega_{R,k} R - \omega_{L,k} R}{b},
\end{align*}
\]

where we denote with: \( x_k \) and \( y_k \), coordinates of the center of axle (mm); \( D_k \), travelled distance between time steps \( k \) and \( k+1 \) (mm); \( v_{t,k} \), mobile robot translation speed (mm/s); \( T \), sampling time (s); \( \Theta_k \), angle between the vehicle and x-axis (°); \( \Delta \Theta_k \), rotation angle between time steps \( k \) and \( k+1 \) (°); \( v_{L,k} \) and \( v_{R,k} \), velocities of the left and right wheel, respectively (mm/s); \( \omega_{L,k} \) and \( \omega_{R,k} \), angular velocities of the left and right wheel, respectively (rad/s); \( R \), radius of the two drive wheels (mm); and \( b \), vehicle axle length (mm). It is assumed that both drive wheels have equal radius. Sampling time \( T \) was 0.1(s). In order to compensate the systematic error regarding the unacquaintance of the exact wheel radius and axle length we expand (6) and (7) with three additional parameters:

\[
\begin{align*}
    v_{t,k} &= \frac{k_1 \cdot v_{L,k} + k_2 \cdot v_{R,k}}{2}, \\
    \omega_k &= \frac{k_2 \cdot v_{R,k} - k_1 \cdot v_{L,k}}{k_3 \cdot b},
\end{align*}
\]

where parameters \( k_1 \) and \( k_2 \) compensate the unacquaintance of the exact wheel radius and parameter \( k_3 \) the unacquaintance of the exact axle length. Detailed explanation of used systematic error compensation and parameter value determination can be found in [4].

Equations (1) to (7) describe the basic odometry pose tracking model and results obtained by simply propagating these equations through time are marked as uncalibrated odometry (UO). Replacing (6) and (7) with (8) and (9) yields the calibrated odometry (CO), which is also used as the motion model for both Kalman filters.

III. KALMAN FILTER BASED POSE TRACKING

The challenge of mobile robot localization using sensor fusion is to weigh its pose (i.e. robot’s state) and sonar range reading (i.e. robot’s output) uncertainty to get the optimal estimate of the pose, i.e. to minimize its covariance. The Kalman filter [10] assumes the Gaussian probability distributions of the state random variable such that it is completely described with the mean and covariance. The optimal state estimate is computed in two major stages: time-update and measurement-update. In the time-update, state prediction is computed on the base of its preceding value and the control input value using the motion model. Measurement-update uses the results from time-update to compute the output predictions with the measurement model. Then the predicted state mean and covariance are corrected in the sense of minimizing the state covariance with the weighted difference between predicted and measured outputs. In succession, motion and measurement models needed for the mobile robot sensor fusion are discussed, and then EKF and UKF algorithms for mobile robot pose tracking are presented.

A. The motion model

The motion model represents the way in which the current state follows from the previous one. State vector is expressed as the mobile robot pose, \( x_k = [x_k \ y_k \ \Theta_k]^T \), with respect to
to a global coordinate frame, where $k$ denotes the sampling instant. Its distribution is assumed to be Gaussian, such that the state random variable is completely determined with a $3 \times 3$ covariance matrix $P_k$ and the state expectation (mean, estimate are used as synonyms). Control input, $u_k$, represents the movement commands which are acted upon the mobile robot to move it from time step $k$ to $k+1$. In the motion model $u_k = [D_k \Delta \Theta_k]^T$ represents translation through distance $D_k$ followed by a rotation through angle $\Delta \Theta_k$. The state transition function $f(\cdot)$ uses the state vector at the current time instant and the current control input to compute the state vector at the next time instant:

$$x_{k+1} = f(x_k, u_k, v_k), \quad (10)$$

where $v_k = [v_{1,k} \ v_{2,k}]^T$ represents unpredictable process noise, that is assumed to be Gaussian with zero mean, $(\mathbb{E}\{v_k\} = [0 \ 0]^T)$, and covariance $Q_k$. With $\mathbb{E}\{\cdot\}$ expectation function is denoted. Using (1) to (3) the state transition function becomes:

$$f(x_k, u_k, v_k) = \begin{bmatrix} x_k + (D_k + v_{1,k}) \cdot \cos(\Theta_k + \Delta \Theta_k + v_{2,k}) \\ y_k + (D_k + v_{1,k}) \cdot \sin(\Theta_k + \Delta \Theta_k + v_{2,k}) \end{bmatrix}, \quad (11)$$

The process noise covariance $Q_k$ was modelled on the assumption of two independent sources of error, translational and angular, i.e. $D_k$ and $\Delta \Theta_k$ are added with corresponding uncertainties. The expression for $Q_k$ is:

$$Q_k = \begin{bmatrix} \sigma_D^2 & 0 \\ 0 & \sigma_{\Theta k}^2 \end{bmatrix}, \quad (12)$$

where $\sigma_D^2$ and $\sigma_{\Theta k}^2$ are variances of $D_k$ and $\Delta \Theta_k$, respectively.

B. The measurement model

The measurement model computes the range between an obstacle and the mobile robot’s center of axle according to a measurement function [7]:

$$h_i(x, p_i) = \sqrt{(x_i - x)^2 + (y_i - y)^2}, \quad (13)$$

where $p_i = (x_i, y_i)$ denotes the point (occupied cell) in the world model detected by the $i$th sonar. The sonar model uses (13) to relate a range reading to the obstacle that caused it:

$$z_{i,k} = h_i(x_k, p_i) + w_{i,k}, \quad (14)$$

where $w_{i,k}$ represents the measurement noise (Gaussian with zero mean and variance $r_{i,k}$) for the $i$th range reading. All range readings are used in parallel, such that range measurements $z_{i,k}$ are simply stacked into a single measurement vector $z_k$. Measurement covariance matrix $R_k$ is a diagonal matrix with the elements $r_{i,k}$. It is to be noted that the measurement noise is additive, which will be beneficial for UKF implementation.

C. EKF based pose tracking

EKF is the first sensor fusion based mobile robot pose tracking technique presented in this paper. Detailed explanation of used EKF localization can be found in [11] and in the sequel only basic equations are presented. Values of the control input vector $u_{k-1}$ computed from wheels’ encoder data are passed to the algorithm at time $k$ such that first time-update is performed obtaining the prediction estimates, and then if new sonar readings are available those predictions are corrected. Predicted (prior) state mean $\hat{x}_k$ is computed in single-shot by propagating the state estimated at instant $k-1$, $\hat{x}_{k-1}$ through the true nonlinear odometry mapping:

$$\hat{x}_k = f(\hat{x}_{k-1}, u_{k-1}, \mathbb{E}\{v_{k-1}\}). \quad (15)$$

The covariance of the predicted state $P_k^{-}$ is approximated with the covariance of the state propagated through a linearized system from (10):

$$P_k^{-} = \nabla f_x P_{k-1}^{-} \nabla f_x^T + \nabla f_u Q_k \nabla f_u^T, \quad (16)$$

where $\nabla f_x = \nabla f_x(\hat{x}_{k-1}, u_{k-1}, \mathbb{E}\{v_{k-1}\})$ is the Jacobian matrix of $f$ with respect to $x$, while $\nabla f_u = \nabla f_u(\hat{x}_{k-1}, u_{k-1}, \mathbb{E}\{v_{k-1}\})$ is the Jacobian matrix of $f(\cdot)$ with respect to control input $u$. It is to be noticed that using (15) and (16) the mean and covariance are accurate only to the first-order of the corresponding Taylor series expansion [12]. If there are no new sonar readings at instant $k$ or if they are all rejected, measurement update does not occur and the state mean and covariance are assigned with the predicted ones:

$$\hat{x}_k = \hat{x}_k, \quad P_k = P_k^{-}. \quad (17) \quad (18)$$

Otherwise, measurement-update takes place where first predictions of the accepted sonar readings are collected in $\hat{z}_k$ with ith component of it being:

$$\hat{z}_{i,k} = h_i(\hat{x}_k, p_i) + E\{w_{i,k}\}. \quad (19)$$

The state estimate and its covariance in time step $k$ are computed as follows:

$$\hat{x}_k = \hat{x}_k^{-} + K_k(z_k - \hat{z}_k^{-}), \quad (20)$$

$$P_k = (I - K_k \nabla h_x)P_k^{-}, \quad (21)$$

where $z_k$ are real sonar readings, $\nabla h_x = \nabla h_x(\hat{x}_k, \mathbb{E}\{w_k\})$ is the Jacobian matrix of the measurement function with respect to the predicted state, and $K_k$ is the optimal Kalman gain computed as follows:

$$K_k = P_k^{-} \nabla h_x^T (\nabla h_x P_k^{-} \nabla h_x^T + R_k)^{-1}. \quad (22)$$

D. UKF based pose tracking

The second sensor fusion based mobile robot pose tracking technique presented in this paper uses UKF. UKF was first proposed by Julier et al. [13], and further developed by Wan and van der Merwe [12]. It utilizes the unscented transformation [13] that approximates the true mean and covariance of a Gaussian random variable propagated through nonlinear
mapping accurate to the inclusively third order of Taylor series expansion for any mapping. Following this, UKF approximates state and output mean and covariance more accurately than EKF and thus superior operation of UKF compared to EKF is expected. UKF was already used for mobile robot localization in [14] to fuse several sources of observations, and the estimates were, if necessary, corrected using interval analysis on sonar measurements. Here we use sonar measurements within UKF, without any other sensors except the encoders to capture angular velocities of the drive wheels (motion model inputs), and without any additional estimate corrections.

Means and covariances are in UKF case computed by propagating carefully chosen so-called pre-sigma points through the true nonlinear mapping. Nonlinear state-update propagating carefully chosen so-called pre-sigma points, the wheels (motion model inputs), and without any additional means and covariances are in UKF case computed by augmenting with means of process noise \( \mathbb{E}\{\nu_{k-1}\} \) only, thus forming extended state vector \( \hat{x}_{k-1}^x \):

\[
\hat{x}_{k-1}^x = \mathbb{E}[x_{k-1}^x] = \left[ \hat{x}_{k-1}^T \mathbb{E}\{v_{k-1}\}^T \right]^T. \quad (23)
\]

Measurement noise does not have to enter the state-update function obtaining matrix \( \mathbf{P}_{k-1}^a \):

\[
\mathbf{P}_{k-1}^a = \begin{bmatrix} \mathbf{P}_{k-1} & 0 \\ 0 & \mathbf{Q}_{k-1} \end{bmatrix}. \quad (24)
\]

Time-update algorithm in time instant \( k \) first requires square root of the \( \mathbf{P}_{k-1}^a \) (or lower triangular Cholesky factorization), \( \sqrt{\mathbf{P}_{k-1}^a} \). Obtained lower triangular matrix is scaled by the factor \( \gamma \):

\[
\gamma = \sqrt{L + \lambda}, \quad (25)
\]

where \( L \) represents the dimension of augmented state \( x_{k-1}^a \) (\( L = 5 \) in this application), and \( \lambda \) is a scaling parameter computed as follows:

\[
\lambda = \alpha^2(L + \kappa) - L. \quad (26)
\]

Parameter \( \alpha \) can be chosen within range \([10^{-4}, 1] \), and \( \kappa \) is usually set to 1. There are 2\( L + 1 \) pre-sigma points, the first is \( \hat{x}_{k-1}^x \) itself, and other 2\( L \) are obtained by adding to or subtracting from \( \hat{x}_{k-1}^x \) each of \( L \) columns of \( \gamma \sqrt{\mathbf{P}_{k-1}^a} \), symbolically written as:

\[
\mathcal{X}_{k-1}^x = \left[ \begin{array}{c} \hat{x}_{k-1}^x \\ \hat{x}_{k-1}^x + \gamma \sqrt{\mathbf{P}_{k-1}^a} \\ \hat{x}_{k-1}^x - \gamma \sqrt{\mathbf{P}_{k-1}^a} \end{array} \right], \quad (27)
\]

where \( \mathcal{X}_{k-1}^x \) represents the matrix whose columns are pre-sigma points. All pre-sigma points are processed by the state-update function obtaining matrix \( \mathcal{X}_{k|k-1}^x \) of predicted states for each pre-sigma point, symbolically written as:

\[
\mathcal{X}_{k|k-1}^x = f[\mathcal{X}_{k-1}^x, \mathbf{u}_{k-1}, \mathcal{X}_{k-1}^u]. \quad (28)
\]

Prior mean is calculated as weighted sum of acquired points:

\[
\hat{x}_{k}^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}_{i,k|k-1}, \quad (29)
\]

where \( \mathcal{X}_{i,k|k-1} \) denotes the \( i \)-th column of \( \mathcal{X}_{k|k-1}^x \). Weights for mean calculation \( W_i^{(m)} \) are given by

\[
W_0^{(m)} = \frac{\lambda}{\lambda + 1}, \quad W_i^{(m)} = \frac{1}{2(L+\lambda)}, \quad i = 1, \ldots, 2L. \quad (30)
\]

Prior covariance matrix \( \mathbf{P}_{k}^- \) is given by

\[
\mathbf{P}_{k}^- = \sum_{i=0}^{2L} W_i^{(m)} \mathbf{Z}_{i,k|k-1} \mathbf{Z}_{i,k|k-1}^T, \quad (31)
\]

where \( \mathbf{W}_i^{(c)} \) represent the weights for covariance calculation which are given by

\[
W_0^{(c)} = \frac{\lambda}{\lambda + 1} + (1 - \alpha^2 + \beta), \quad W_i^{(c)} = \frac{1}{2(L+\lambda)}, \quad i = 1, \ldots, 2L. \quad (32)
\]

For Gaussian distributions \( \beta = 2 \) is optimal. If there are new sonar readings available at time instant \( k \), predicted readings of accepted sonars for each sigma-point are grouped in matrix \( \mathbf{Z}_{k|k-1} \) obtained by

\[
\mathbf{Z}_{k|k-1} = h[\mathcal{X}_{k|k-1}^x, \mathbf{p}] + \mathbb{E}\{\mathbf{w}_k\}, \quad (33)
\]

where \( \mathbf{p} \) denotes the series of points in the world map predicted to be hit by sonar beams. Predicted readings \( \hat{z}_{k}^- \) are then

\[
\hat{z}_{k}^- = \sum_{i=0}^{2L} W_i^{(m)} \mathbf{Z}_{i,k|k-1}. \quad (34)
\]

To prevent the sonar readings that hit near the corner of obstacles to influence on the measurement correction, since their probability distribution cannot be approximated with Gaussian, another threshold comparison were made. These problematic sonar readings are recognized with mean \( \hat{z}_{k}^- \) that differs from \( z_{i,k} \) more than the acceptance threshold amounts, and those are being discarded. Readings covariance is

\[
\mathbf{P}_{zz} = \sum_{i=0}^{2L} W_i^{(c)}[\mathbf{Z}_{i,k|k-1} - \hat{z}_{k}^-][\mathbf{Z}_{i,k|k-1} - \hat{z}_{k}^-]^T + \mathbf{R}_k, \quad (35)
\]

and state-output cross-covariance matrix is

\[
\mathbf{P}_{z_x} = \sum_{i=0}^{2L} W_i^{(c)}[\mathcal{X}_{i,k|k-1} - \hat{x}_{k}^-][\mathbf{Z}_{i,k|k-1} - \hat{z}_{k}^-]^T. \quad (36)
\]

Kalman gain \( \mathbf{K}_k \) is

\[
\mathbf{K}_k = \mathbf{P}_{z_x} \mathbf{P}_{zz}^{-1}. \quad (37)
\]

Posterior state covariance is finally calculated as

\[
\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{zz} \mathbf{K}_k^T. \quad (38)
\]

The measurement correction is done as in (20).
IV. EXPERIMENTAL RESULTS

The described pose tracking techniques are tested in parallel on a Pioneer 2DX mobile robot from ActivMedia Robotics. Two series of experiments were made to evaluate described localization approaches. The first one was performed in our department corridor, and the second one in a larger room of our department. Main characteristic of the first experiment world, shown in Fig. 3, is that it has little features along the axis through the corridor middle and the used occupancy grid model is accurate. The mobile robot starts in the right corridor end and travels to the left end. Main characteristic of the second one, shown in Fig. 4, is that it has enough features along both axis but used occupancy grid model is less accurate because this room is full of easily movable furniture (like chairs, tables, trash baskets, etc.) which never stays for long time in the same place. In this experiment setup the mobile robot starts in the left room and has to reach the right room. This two setups enable us to test the local localization algorithm performance in an environment with little features and in a badly/incompletely modelled environment. Traversed paths for both experiment setups (Fig. 3 and 4) are presented by a line with arrows showing mobile robot motion succession. A gradient navigation module [15] is used for mobile robot control, i.e. for path planning and local obstacles avoidance. Regarding navigation module’s mobile robot pose input, two experiments in each of the environments were made. For navigation module pose input, EKF pose estimation is used in the first one and UKF in the second one. New sonars range measurements $z_k$ are available every three time steps on this mobile robot. Raw sonar data $z_k$ and predicted readings $h_k$ differ because the real mobile robot pose is not exactly known and due to sonar measurement noise, occlusions, specular reflections, outliers, and used occupancy grid model inaccuracy. So $z_k$ are first compared to $h_k$ and only those readings with difference under a certain threshold are accepted. Used threshold is set to 5 cells. That means, in a grid map with cell size of $100(mm) \times 100(mm)$, as in our case, readings with difference less then $500(mm)$ are accepted. Note that this procedure is repeated once more in UKF case when prediction $\hat{z}_k^p$ is obtained. Measurement variances of $\Delta \theta$ and $D$ (process noise) are set to 25 and 100 respectively and measurement noise variance value is set to $R_k = R = 10^0 \cdot I$.

Obtained results are presented in Fig. 5 and 6 for the first experiment setup and in Fig. 7 and 8 for the second experiment setup. Exact final mobile robot pose was manually measured at experiment end. In presented figures robot position part is denoted by a black cross and the orientation part by a short thick line. Estimated final position for each tested technique is denoted by a black dot. Table I summaries the results of all performed experiments.

<table>
<thead>
<tr>
<th></th>
<th>EKF</th>
<th>UKF</th>
<th>CO</th>
<th>UO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position error (%)</td>
<td>1.53</td>
<td>1.10</td>
<td>3.71</td>
<td>11.07</td>
</tr>
<tr>
<td>Orientation error (%)</td>
<td>2.08</td>
<td>3.11</td>
<td>4.72</td>
<td>10.71</td>
</tr>
<tr>
<td>Average error (%)</td>
<td>1.81</td>
<td>2.11</td>
<td>4.22</td>
<td>10.89</td>
</tr>
</tbody>
</table>

Position error is calculated as:

$$\text{Position Error} = \frac{\text{Pos}_{\text{act}} - \text{Pos}_{\text{est}}}{\text{Dist}} \cdot 100\%,$$

where $\text{Pos}_{\text{act}}$ is actual final position, $\text{Pos}_{\text{est}}$ estimated final position and $\text{Dist}$ total distance traversed by the mobile robot. Orientation error is calculated as:

$$\text{Orientation Error} = \frac{\text{Orient}_{\text{act}} - \text{Orient}_{\text{est}}}{\text{Orient}_{\text{act}}} \cdot 100\%,$$

where $\text{Orient}_{\text{act}}$ is actual final orientation and $\text{Orient}_{\text{est}}$ estimated final orientation. Average error is calculated as the average value between the position and orientation error.

V. CONCLUSION

Two mobile robot pose tracking techniques based on Kalman filter theory and occupancy grid maps are implemented and experimentally compared: extended and unscented Kalman filter. It is shown that odometry calibration and sensor fusion can significantly improve mobile robot pose tracking i.e. local localization. Both here presented sensor fusion approaches can correct the mobile robot position so that the error bound lies inside one occupancy grid map cell, which is the best possible considering used world model. The UKF approach results in a smooth mobile robot pose estimate while the EKF approach produces bigger correction values thus reacting faster to increased real and estimated mobile robot pose discrepancy. EKF approach is somewhat computationally simpler because it predicts the sonar range measurements only once. UKF approach performs better.
in an accurately modelled environment. Both approaches can cope with an environment with little features and a badly modelled environment. Average pose error somewhat increases in badly modelled environment for about 2 percent. Future work on this topic will include a dual estimation of both robot pose and calibration parameters in kinematic model.

REFERENCES