Extended Kalman filter based mobile robot pose tracking using occupancy grid maps

Conference Paper · June 2004

DOI: 10.1109/MELCON.2004.1346851 · Source: IEEE Xplore

2 authors:

Edouard Ivanjko
University of Zagreb
85 PUBLICATIONS **444** CITATIONS
SEE PROFILE

Ivan Petrovic
University of Zagreb
267 PUBLICATIONS **3,019** CITATIONS
SEE PROFILE

Extended Kalman Filter based Mobile Robot Pose Tracking using Occupancy Grid Maps

E. Ivanjko, I. Petrović
University of Zagreb/Faculty of Electrical Engineering and Computing, Zagreb, Croatia
edouard.ivanjko@fer.hr, ivan.petrovic@fer.hr

***Abstract*— Mobile robot pose tracking is mostly based on odometry. However, with time, odometric pose tracking accumulates errors in an unbounded fashion. This paper describes a way to decrease the odometry error by using an Extended Kalman Filter (EKF) for fusion of calibrated odometry data and sonar readings. Common approaches for calibrated odometry and sonar fusion use a feature based map which has two uncertainties in the measurement process. One uncertainty is related to the sonar range reading and the other one to the feature/range reading assignment. Our approach is adapted to an occupancy grid map which has only the sonar range reading uncertainty in the measurement process. Experimental results on the mobile robot Pioneer 2DX show improved accuracy of the pose estimation compared to the calibrated odometry.**

I. Introduction

Mobile robot pose tracking is one of the very important tasks in navigation of autonomous mobile robots [1]. In a typical indoor environment with a flat floor plan, pose tracking becomes a matter of estimating the Cartesian coordinates (x, y) of the robot (robot position) and its orientation Θ. Odometry is one of the most important means of achieving this task. 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, discretised 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 absolute pose updates using additional sensors [2].

A lot of research have 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. First part consists of a better error [3] or odometry model [4] and second part consists of using additional sensors [2], [5]. 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 or for online odometry calibration [6]. In case of mobile robot pose correction local and global world model matching [7] or sensor fusion techniques are used [8], [9].

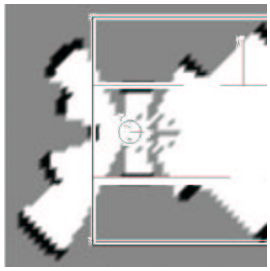

Fig. 1. Local occupancy grid map presenting the bad sonar placement influence.

Our approach is based on sensor fusion with an Extended Kalman Filter (EKF). The idea is to match recent sensory information against prior knowledge of the environment, i.e. a world model. Due to problems with bad sonar placement on our mobile robot, a sonar based local map with high quality can't be determined [10]. A local occupancy grid map obtained using our mobile robot is given in Fig 1. It can be noticed that only sonar 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 [8] is adopted that can find direct correspondences between mobile robot sensor readings and sensor readings predicted from a world model. False sonar range readings can easily be rejected by using a threshold comparison between predicted and measured sonar range readings.

II. Mobile Robot Model

Mobile robot used in our experiments is a three-wheeled robot. Two front wheels are drive wheels with encoder mounted on them and the third wheel is a castor wheel needed for robot stability. Drive wheels can be controlled independently from each other. The kinematic model of the mobile robot is given by the following relations [2]:

$$x(k+1)=x(k)+D(k)\cdot\cos\Theta(k+1), \quad (1)$$

$$y(k+1)=y(k)+D(k)\cdot\sin\Theta(k+1), \quad (2)$$

$$\Theta(k+1)=\Theta(k)+\Delta\Theta(k), \quad (3)$$

Acknowledgment

This research has been supported by the Ministry of Science and Technology of the Republic of Croatia under grant No. 0036018.

$$D(k)=v_t(k)\cdot T \tag{4}$$

$$\Delta\Theta(k)=\omega(k)\cdot T \tag{5}$$

$$v_t(k)=\frac{v_L(k)+v_R(k)}{2}=\frac{\omega_L(k)R+\omega_R(k)R}{2}, \tag{6}$$

$$\omega(k)=\frac{v_R(k)-v_L(k)}{b}=\frac{\omega_R(k)R-\omega_L(k)R}{b}, \tag{7}$$

where are: $x(k)$ and $y(k)$ coordinates of the center of axle (mm); $D(k)$ travelled distance between time step k and $k+1$ (mm); $v_t(k)$ 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 step 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 the unacquaintance of the exact axle length we expand the equations (6) and (7) with three additional parameters:

$$v_t(k)=\frac{k_1\cdot v_L(k)+k_2\cdot v_R(k)}{2}, \tag{8}$$

$$\omega(k)=\frac{k_2\cdot v_R(k)-k_1\cdot v_L(k)}{k_3\cdot b}, \tag{9}$$

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 using this technique are marked as uncalibrated odometry (UO). Replacing (6) and (7) with (8) and (9) gives the calibrated odometry (CO) pose tracking model as the second mobile robot pose tracking technique that we compared with our EKF approach.

III. EKF BASED SENSOR FUSION

The Kalman Filter is a recursive solution to the discrete-data linear filtering problem and is very powerful in estimation of past, present and future states even if the precise nature of the modelled system is unknown [11]. In case of a non-linear process an extension of the original Kalman Filter is used. It linearizes the non-linear process about the current mean and covariance. It's referred to as an Extended Kalman Filter or EKF. Both approaches are common practice in mobile robotics [2], [5], [8].

The challenge of mobile robot pose tracking is to weigh the pose and sonar range reading uncertainty to get the best estimate of the mobile robot's pose. The Extended Kalman Filter is here used as a probabilistic tool to do this task i.e. to extract the best estimate from multiple sources of information corrupted by Gaussian noise.

Extended Kalman Filter uses two models to characterize the behavior of the mobile robot. That's the plant model and measurement model. The plant model consists of aforementioned mobile robot model and uncertainty tracking part. The measurement model uses knowledge of the mobile robot's pose and environmental model to predict the range readings from the mobile robot's range sensors (in our case sonar). It also calculates the uncertainty associated with each reading. The environmental model is usually feature based [8], but in our approach it is based on an occupancy grid map [12]. Both EKF models are described in succession.

A. The Plant Model

The plant model represents the way in which the current state (in our case the mobile robot pose) is derived form the previous state. State vector is expressed as the mobile robot pose, $\mathbf{x}(k) = [x(k), y(k), \Theta(k)]^T$, with respect to a global coordinate frame. Each state vector $\mathbf{x}(k)$ has a degree of uncertainty which is represented as a 3 by 3 covariance matrix $\mathbf{P}(k)$. A more detailed notation, $\mathbf{P}(k+1|k)$, is used to represent the covariance of state $\mathbf{x}(k+1)$ given all the sensory information up to and including time step k. The idea is to decrease the uncertainty about $\mathbf{x}(k+1)$ using sensor information gathered at time step $k+1$. Control input, $\mathbf{u}(k)$, represents the movement commands which are acted upon by the mobile robot to move it from time step k to $k+1$. In our implementation control input, $\mathbf{u}(k) = [D(k), \Delta\Theta(k)]^T$, represents rotation through angle $\Delta\Theta(k)$ followed by a translation through distance $D(k)$. Control input values are computed using velocities of right and left wheels. The state transition function, $\mathbf{f}(\cdot)$, uses the state vector and control input to compute the state vector at the next time step:

$$\mathbf{x}(k+1|k)=\mathbf{f}(\mathbf{x}(k|k),\mathbf{u}(k),E\{\mathbf{v}(k)\}), \tag{10}$$

where $\mathbf{v}(k)$ represents unpredictable noise. The noise is assumed to be Gaussian with zero mean, ($E\{\mathbf{v}(k)\} = 0$), and covariance $\mathbf{Q}(k)$. From the equations (1) to (3) the state transition function becomes:

$$\mathbf{f}(\mathbf{x}(k),\mathbf{u}(k),0)=\begin{bmatrix} x(k)+D(k)\cdot\cos(\Theta(k)+\Delta\Theta(k)) \\ y(k)+D(k)\cdot\sin(\Theta(k)+\Delta\Theta(k)) \\ \Theta(k)+\Delta\Theta(k) \end{bmatrix}. \tag{11}$$

The noise covariance, $\mathbf{Q}(k)$, was modelled on the assumption that there are two independent sources of error, angular and translational. Whence $\mathbf{x}(k+1|k)$ depends upon $\mathbf{x}(k|k)$, $\Delta\Theta(k)$ and $D(k)$, we must translate the uncertainty in $\Delta\Theta(k)$ and $D(k)$ into uncertainty in $\mathbf{x}(k+1|k)$. This is done by partial differentiation of (11) with respect to $\Delta\Theta(k)$ and $D(k)$, what gives the following Jacobian:

$$\nabla\mathbf{f}(k)=\begin{bmatrix} -D(k)\cdot\sin(\Theta(k)+\Delta\Theta(k) & \cos(\Theta(k)+\Delta\Theta(k) \\ D(k)\cdot\cos(\Theta(k)+\Delta\Theta(k) & \sin(\Theta(k)+\Delta\Theta(k) \\ 1 & 0 \end{bmatrix}. \tag{12}$$

The complete expression for $\mathbf{Q}(k)$ is then:

$$\mathbf{Q}(k)=\nabla\mathbf{f}(k)\begin{bmatrix} \Delta\Theta(k)^2\sigma^2_{\Delta\Theta} & 0 \\ 0 & \sigma^2_D \end{bmatrix}\nabla\mathbf{f}(k)^T, \tag{13}$$

where $\sigma^2_{\Delta\Theta}$ and σ^2_D are variances of $\Delta\Theta(k)$ and D.

Another source of uncertainty is the uncertainty in the position and orientation at time step k, $\mathbf{P}(k|k)$, carried

forward to time step $k+1$. So we need an another Jacobian to determine how the uncertainty is transferred between the time steps. To compute it we must make an another partial differentiation of (11) with respect to $x(k)$, $y(k)$ and $\theta(k)$. The resulting Jacobian, $\nabla\mathbf{f}'(k)$, is given by:

$$\nabla\mathbf{f}'(k)=\begin{bmatrix} 1 & 0 & -D(k)\cdot\sin(\Theta(k)+\Delta\Theta(k)) \\ 0 & 1 & D(k)\cdot\cos(\Theta(k)+\Delta\Theta(k)) \\ 0 & 0 & 1 \end{bmatrix}, \quad (14)$$

and thus the complete pre-localization covariance matrix expression for $\mathbf{x}(k+1|k)$ becomes:

$$\mathbf{P}(k+1|k)=\nabla\mathbf{f}'(k)\cdot\mathbf{P}(k|k)\cdot\nabla\mathbf{f}'(k)^T+\mathbf{Q}(k). \quad (15)$$

B. The Measurement Model

Sonar range readings are used to improve the mobile robot pose estimation given by (10) and (11). To compute the range between a detected obstacle and the mobile robot we define a measurement function [8]:

$$h_i(x(k+1|k),p_i)=\sqrt{(x_i-x(k+1|k))^2+(y_i-y(k+1|k))^2}, \quad (16)$$

where $p_i=(x_i,y_i)$ denotes the point (occupied cell) in the world model detected by the ith sonar. The sonar model uses (16) to relate a range reading to the obstacle that caused it:

$$z_i(k)=h_i(x(k+1|k),p_i)+w_i(k), \quad (17)$$

where w_i represents the measurement noise for the ith accepted range reading. It's assumed that the noise is Gaussian with zero mean and variance $r_i(k)$. Except noise in the sensing process, unacquaintance of exact mobile robot pose in the world model can cause difference between real and predicted range readings. To include this error we must translate the mobile robot pose uncertainty into measurement uncertainty using the measurement Jacobian $\nabla\mathbf{h}_i$. The Jacobian is derived from (16) by partial differentiation of $h_i(k)$ with respect to $x(k+1|k)$, $y(k+1|k)$ and $\theta(k+1|k)$:

$$\nabla\mathbf{h}_i=\frac{1}{\sqrt{(x_i-x(k+1|k))^2+(y_i-y(k+1|k))^2}}\begin{bmatrix} x(k+1|k)-x_i \\ y(k+1|k)-y_i \\ 0 \end{bmatrix}^T. \quad (18)$$

Using the above equations we can now define the EKF equations for the mobile robot pose correction [8], [11]. All range readings are used in parallel, range measurements $z_i(k)$ are simply stacked into a single measurement vector $\mathbf{z}(k)$. The same approach is used for the rest of the described variables. Innovation covariance matrix, $\mathbf{S}(k)$, can be computed as:

$$\mathbf{S}(k+1)=\nabla\mathbf{h}\cdot\mathbf{P}(k+1|k)\cdot\nabla\mathbf{h}^T+\mathbf{R}(k+1), \quad (19)$$

where $\mathbf{R}(k)$ denotes the measurement noise matrix. It's a diagonal matrix with the $r_i(k+1)$ values on the diagonal. Kalman gain can now be computed as follows:

$$\mathbf{K}(k+1)=\mathbf{P}(k+1|k)\cdot\nabla\mathbf{h}^T\cdot\mathbf{S}^{-1}(k+1). \quad (20)$$

The variance of the new pose estimate is computed as:

$$\mathbf{P}(k+1|k+1)=\mathbf{P}(k+1|k)-\mathbf{K}(k+1)\cdot\mathbf{S}(k+1)\cdot\mathbf{K}^T(k+1). \quad (21)$$

The form of the EKF to take the latest range readings for the new pose estimate is now:

$$\mathbf{x}(k+1|k+1)=\mathbf{x}(k+1|k)+\mathbf{K}(k+1)\cdot(\mathbf{z}(k+1)-\mathbf{h}(k+1)). \quad (22)$$

IV. Experimental results

The described pose tracking technique was tested on a Pioneer 2DX mobile robot from ActivMedia Robotics. New sonars' range measurements $\mathbf{z}(k)$ are available every three time steps on this mobile robot. These readings are used in (22) to correct the predicted mobile robot pose with (10). To compute the pose correction sonars' range readings are compared to range readings predicted using the world model $\mathbf{h}(k)$, according to (16). In an ideal case raw sonar data $\mathbf{z}(k)$ and predicted readings $\mathbf{h}(k)$ would be the same. In a real case they differ because the real mobile robot pose is not exactly known and due to sonars' measurement noise, occlusions, specular reflections and outliers. So $\mathbf{z}(k)$ were first compared to $\mathbf{h}(k)$ and only those readings which difference were under a certain threshold were accepted. Used threshold was set to 3 cells. That means, in a grid map with cell size of $100(mm)\cdot 100(mm)$ readings with difference less then $300(mm)$ were accepted. $\Delta\Theta(k)$ and D measurement variances were set to 25 and 10 respectively and measurement noise variance value was set to $\mathbf{R}=10\cdot\mathbf{I}$.

We performed two experiments in the corridor of our department. The main feature of this test world is that it has a very small number of features in the x axis. The first experiment was a straight line motion where the navigation strategy was to hold the mobile robot orientation near $180(^\circ)$. Travelled distance was $10(m)$. The second experiment was performed using a gradient navigation module [13]. Travelled distance was $20(m)$ in this case. In both experiments the mobile robot start pose was the same. Position and orientation data were recorded and are shown in Figs. 2 to 5. EKF data are presented by a solid line, calibrated odometry (CO) by a dash-dot line, and uncalibrated odometry (UO) by a dashed line. Exact final mobile robot position is denoted by a black cross and the calculated final position for each tested technique by a black dot. Exact final mobile robot orientation is denoted by a short thick line.

Table I summaries the results of performed experiments. Position errors are calculated as:

$$PositionError=\frac{Pos_{act}-Pos_{est}}{Dist}\cdot 100\%, \quad (23)$$

where Pos_{act} is the actual final position, Pos_{est} is the estimated final position and $Dist$ is the total distance traversed by the mobile robot. Orientation errors are calculated as:

$$OrientationError=\frac{Orient_{act}-Orient_{est}}{Orient_{act}}\cdot 100\%, \quad (24)$$

where $Orient_{act}$ is the actual final orientation and $Orient_{est}$ the estimated final orientation. Average errors are calculated as the average value between the position and orientation errors.

From experimental results it is obvious that EKF based approach produces better accuracy than calibrated odometry, particulary in the position estimation.

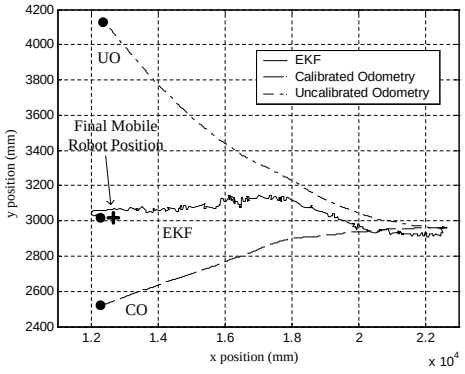

Fig. 2. Position in the straight line experiment.

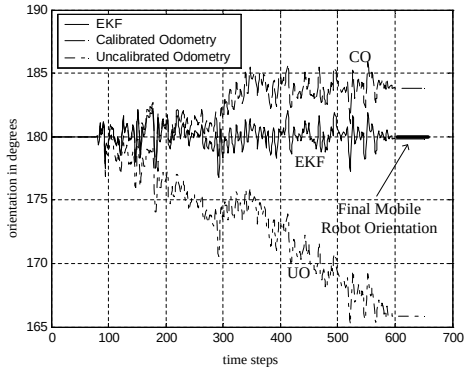

Fig. 3. Orientation in the straight line experiment.

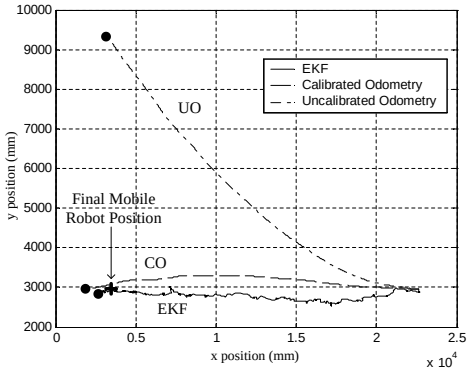

Fig. 4. Position in the gradient navigation experiment.

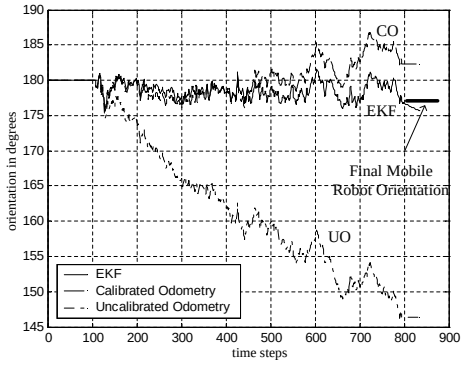

Fig. 5. Orientation in the gradient navigation experiment.

TABLE I
ERROR COMPARISON OF THREE DIFFERENT TECHNIQUES

	EKF	Calibrated Odometry	Uncalibrated Odometry
Position error (%)	4.3	9.6	22.2
Orientation error (%)	4.1	5.5	13.3
Average error (%)	4.2	7.6	17.8

V. CONCLUSION AND FUTURE WORK

An EKF based pose tracking technique has been implemented and experimentally compared to common used pose tracking techniques. It is shown that additional sensors fused with calibrated odometry can improve the mobile robot pose tracking and compensate environmental influence on the mobile robot. Future work on this topic will include the use of sonar readings for on-line odometry calibration.

REFERENCES

[1] J. Borenstein, H. R. Everett, and L. Feng, *Where am I? Sensors and Methods for Mobile Robot Positioning.* Ann Arbor, MI 48109: University of Michigan, 1996.

[2] P. Goel, S. I. Roumeliotis, and G. Sukhatme, "Robust localization using relative and absolute position estimates," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1999.

[3] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Transactions in Robotics and Automation*, vol. 12, 1996.

[4] E. Ivanjko, I. Petrović, and N. Perić, "An approach to odometry calibration of differential drive mobile robots," in *Proc. Internation Conference on Electrical Drives and Power Electronics EDPE'03*, The High Tatras, Slovakia, Sept. 24–26, 2003, pp. 519–523.

[5] I. Petrović, E. Ivanjko, and N. Perić, "Neural network based corrections of odometry errors in mobile robots," in *Proceedings of the FIRA Congress*, Seul, South Korea, May 26–29, 2002, pp. 91–96.

[6] N. Roy and S. Thrun, "Online self-calibration for mobile robots," in *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[7] K. Konolige, "Markov localization using correlation," in *International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999.

[8] D. Lee, *The Map-Building and Exploration Strategies of a Simple Sonar-Equipped Robot*. Trumpington Street, Cambridge CB2 1RP: Cambridge University Press, 1996.

[9] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*. Trumpington Street, Cambridge CB2 1RP: Cambridge University Press, 2000.

[10] E. Ivanjko, I. Petrović, and K. Maček, "Improvements of occupancy grid maps by sonar data corrections," in *Proceedings of 2003 FIRA Robot Soccer World Congress*, Vienna, Austria, Oct. 1–3, 2003.

[11] G. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina at Chapel Hill, NC, Tech. Rep. TR 95-041, Nov. 2000.

[12] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, St. Louis, USA, Mar. 1985, pp. 116–121.

[13] K. Konolige, "A gradient method for realtime robot control," in *Proceedings of International Conference on Intelligent Robots and Systems IROS 2000*, 2000.