Extended Kalman and Particle Filtering for sensor fusion in mobile robot localization

Gerasimos G. Rigatos

***Abstract*— State estimation is a major problem in mobile robot localization. To this end gaussian and nonparametric filters have been developed. In this paper the Extended Kalman Filter which assumes gaussian measurement noise is compared to the Particle Filter which does not make any assumption on the measurement noise distribution. As a case study the estimation of the state vector of a mobile robot is used, when measurements are available from both odometric and sonar sensors. It is shown that in this kind of sensor fusion problem the Particle Filter stands has improved performance and has wider applications than the Extended Kalman Filter, at the cost of more demanding computations.**

I. Introduction

State estimation (or filtering) is a research field of primary importance for industrial systems operation. It is well known that the optimal filter for linear model with Gaussian noise is the Kalman Filter [1]. State estimation for nonlinear systems with non-Gaussian noise is a difficult problem and in general the optimal solution cannot be expressed in closed-form. Suboptimal solutions use some form of approximation such as model linearisation in the Extended Kalman Filter (EKF) [2]. More recently, Monte Carlo sampling from state vectors distribution has been used in the development of the particle filter. A particular advantage of this sample-based approximation is its suitability in applying it to the nonlinear non-Gaussian case [3-6].

The Extended Kalman Filter (EKF) is an incremental estimation algorithm that performs optimization in the least mean squares sense and which has been successfully applied to neural networks training and to data fusion problems [2,7]. In this paper the EKF has been employed for the localization of an autonomous vehicle by fusing data coming from different sensors. In the EKF approach the state vector is approximated by a Gaussian random variable, which is then propagated analytically through the first order linearization of the nonlinear system. The series approximation in the EKF algorithm can, however, lead to poor representations of the nonlinear functions and of the associated probability distributions. As a result, sometimes the filter will be divergent.

To overcome these shortcomings, a new kind of nonlinear filtering method, the so-called Particle Filter (PF), has been proposed [8]. Particle filtering has improved performance over the established nonlinear filtering approaches (e.g. the EKF), since it can provide optimal estimation in nonlinear non-Gaussian state-space models ([9-11]), as well as estimation of nonlinear models ([12-13]). Particle filters can estimate the system states sufficiently when the number of particles (estimations of the state vectors which evolve in parallel) is large. However the method has not yet become popular in industry because implementation details are missing in literature, and because its computational complexity has to be handled in real-time applications [14,15]. The particle filtering algorithm reminds of the genetic algorithms where a number of N particles is subject to a mutation mechanism which corresponds to the prediction stage, and to selection mechanism which corresponds to the correction stage [16].

In this paper implementation and tuning issues of particle filtering are discussed. The performance of the proposed methodology is evaluated against EKF in the problem of sensor fusion for the localization of an autonomous mobile robot. The problem is to succeed an accurate estimation of the state vector of the mobile robot fusing measurements from odometric and sonar sensors. At a second stage the estimated state vector is used by a nonlinear controller in-order to make the mobile robot track a desired trajectory.

The structure of the paper is as follows: In Section II Data Fusion with the use of Extended Kalman Filtering is discussed. The Extended Kalman Filter (EKF) for the Nonlinear state-measurement model is presented. In Section III the Particle Filtering algorithm for state estimation of nonlinear dynamical systems is introduced. Particle filtering based on sequential importance resampling is analyzed. The prediction and correction stages are explained. Issues for improved resampling and substitution of the degenerated particles are discussed. In Section IV simulation experiments are carried out to evaluate the performance of the Extended Kalman Filter and the Particle Filter in sensor fusion for mobile-robot localization. Finally, in Section V concluding remarks are stated.

II. Data Fusion with the use of Extended Kalman Filtering

A. EKF for the Nonlinear State-Measurement Model

The following nonlinear time-invariant state model is now considered [2]:

$$\begin{aligned} x(k+1) &= \phi(x(k)) + w(k) \\ z(k) &= \gamma(x(k)) + v(k) \end{aligned} \tag{1}$$

where $w(k)$ and $v(k)$ are uncorrelated, zero-mean, Gaussian zero-mean noise processes with covariance matrices $Q(k)$ and $R(k)$ respectively. The operators $\phi(x)$ and $\gamma(x)$ are given by, $\phi(x) = [\phi_1(x), \phi_2(x), \cdots, \phi_m(x)]^T$, and $\gamma(x) =$

G. G. Rigatos is with the Unit of Industrial Automation, Industrial Systems Institute, Stadiou str., 26504 Rion Patras, Greece; grigat@isi.gr

$[\gamma_1(x), \gamma_2(x), \cdots, \gamma_p(x)]^T$, respectively. It is assumed that ϕ and γ are sufficiently smooth in x so that each one has a valid series Taylor expansion. Following a linearization procedure, ϕ is expanded into Taylor series about $\hat{x}$:

$$\phi(x(k)) = \phi(\hat{x}(k)) + J_\phi(\hat{x}(k))[x(k) - \hat{x}(k)] \tag{2}$$

where $J_\phi(x)$ is the Jacobian of ϕ calculated at $\hat{x}(k)$:

$$J_\phi(x) = \frac{\partial \phi}{\partial x}|_{x=\hat{x}(k)} = \begin{pmatrix} \frac{\partial \phi_1}{\partial x_1} & \frac{\partial \phi_1}{\partial x_2} & \cdots & \frac{\partial \phi_1}{\partial x_N} \\ \frac{\partial \phi_2}{\partial x_1} & \frac{\partial \phi_2}{\partial x_2} & \cdots & \frac{\partial \phi_2}{\partial x_N} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \phi_k}{\partial x_1} & \frac{\partial \phi_k}{\partial x_2} & \cdots & \frac{\partial \phi_k}{\partial x_N} \end{pmatrix} \tag{3}$$

Likewise, γ is expanded about $\hat{x}^-(k)$

$$\gamma(x(k)) = \gamma(\hat{x}^-(k)) + J_\gamma[x(k) - \hat{x}^-(k)] + \cdots \tag{4}$$

where $\hat{x}^-(k)$ and $\hat{x}(k)$ were defined in sub-section **??**. The Jacobian $J_\gamma(x)$ is

$$J_\gamma(x) = \frac{\partial \gamma}{\partial x}|_{x=\hat{x}^-(k)} = \begin{pmatrix} \frac{\partial \gamma_1}{\partial x_1} & \frac{\partial \gamma_1}{\partial x_2} & \cdots & \frac{\partial \gamma_1}{\partial x_N} \\ \frac{\partial \gamma_2}{\partial x_1} & \frac{\partial \gamma_2}{\partial x_2} & \cdots & \frac{\partial \gamma_2}{\partial x_N} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \gamma_k}{\partial x_1} & \frac{\partial \gamma_k}{\partial x_2} & \cdots & \frac{\partial \gamma_k}{\partial x_N} \end{pmatrix} \tag{5}$$

The resulting expressions create first order approximations of ϕ and γ. Thus the linearized version of the plant is obtained:

$$\begin{array}{l} x(k+1) = \phi(\hat{x}(k)) + J_\phi(\hat{x}(k))[x(k) - \hat{x}(k)] + w(k) \\ z(k) = \gamma(\hat{x}^-(k)) + J_\gamma(\hat{x}^-(k))[x(k) - \hat{x}^-(k)] + v(k) \end{array}$$

Now, the EKF recursion is as follows: First the time update is considered: by $\hat{x}(k)$ the estimation of the state vector at instant k is denoted. Given initial conditions $\hat{x}^-(0)$ and $P^-(0)$ the recursion proceeds as:

- *Measurement update*. Acquire $z(k)$ and compute:

$$\begin{array}{l} K(k) = P^-(k) J_\gamma^T(\hat{x}^-(k)) \cdot \\ \cdot [J_\gamma(\hat{x}^-(k)) P^-(k) J_\gamma^T(\hat{x}^-(k)) + R(k)]^{-1} \\ \hat{x}(k) = \hat{x}^-(k) + K(k)[z(k) - \gamma(\hat{x}^-(k))] \\ P(k) = P^-(k) - K(k) J_\gamma(\hat{x}^-(k)) P^-(k) \end{array} \tag{6}$$

- *Time update*. Compute:

$$\begin{array}{l} P^-(k+1) = J_\phi(\hat{x}(k)) P(k) J_\phi^T(\hat{x}(k)) + Q(k) \\ \hat{x}^-(k+1) = \phi(\hat{x}(k)) \end{array} \tag{7}$$

The schematic diagram of the EKF loop is given in Fig. 1.

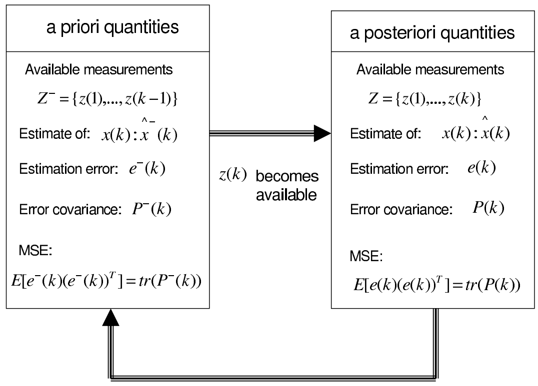


Fig. 1. Schematic diagram of the EKF loop

III. Particle filtering for the nonlinear state-measurement model

A. Particle Filter with sequential importance resampling

In the general case the equations of the optimal filter used for the calculation of the state-vector of a nonlinear dynamical system do not have an explicit solution. This happens for instance when the process noise and the noise of the output measurement do not follow a Gaussian distribution. In that case approximation through Monte-Carlo methods can used. As in the case of the Kalman Filter or the Extended Kalman Filter the particles filter consists of the measurement update (correction stage) and the time update (prediction stage) [17].

1) The prediction stage: The prediction stage calculates $p(x(k)|Z^-)$ where $Z^- = \{z(1), \cdots, z(n-1)\}$, using:

$$p(x(k-1)|Z^-) = \sum_{i=1}^{N} w_{k-1}^i \delta_{\xi_{k-1}^i}(x(k-1)) \tag{8}$$

while from Bayes formula it holds $p(x(k)|Z^-) = \int p(x(k)|x(k-1))p(x(k-1)|Z^-)dx$. This finally gives

$$\begin{array}{c} p(x(k)|Z^-) = \sum_{i=1}^{N} w_{k-1}^i \delta_{\xi_{k^-}^i}(x(k)) \\ \text{with} \quad \xi_{k^-}^i \sim p(x(k)|x(k-1) = \xi_{k-1}^i) \end{array} \tag{9}$$

The meaning of Eq. (9) is as follows: the state equation of the nonlinear system of Eq. (1) is executed N times, starting from the N previous values of the state vectors $x(k-1) = \xi_{k-1}^i$ and using Eq. (1). This means that the value of the state vector which is calculated in the prediction stage is the result of the weighted averaging of the state vectors which were calculated after running the state equation, starting from the N previous values of the state vectors ξ_{k-1}^i.

2) The correction stage: The a-posteriori probability density was performed using Eq. (9). Now a new position measurement $z(k)$ is obtained and the objective is to calculate the corrected probability density $p(x(k)|Z)$, where $Z = \{z(1), z(2), , z(k)\}$. From Bayes law it holds that $p(x(k)|Z) = \frac{p(Z|x(k))p(x(k))}{p(Z)}$, which finally results into

$$p(x(k)|Z) = \sum_{i=1}^{N} w_k^i \delta_{\xi_{k-}^i}(x(k))$$
$$\text{where } w_k^i = \frac{w_{k-}^i p(z(k)|x(k)=\xi_{k-}^i)}{\sum_{j=1}^{N} w_{k-}^j p(z(k)|x(k)=\xi_{k-}^j)} \tag{10}$$

Eq. (10) denotes the corrected value for the state vector. The recursion of the Particle Filter proceeds in a way similar to the update of the Kalman Filter or the Extended Kalman Filter, i.e.:

Measurement update: Acquire $z(k)$ and compute the new value of the state vector

$$p(x(k)|Z) = \sum_{i=1}^{N} w_k^i \delta_{\xi_{k-}^i}(x(k))$$

$$\text{with corrected weights } \ w_k^i = \frac{w_{k-}^i p(z(k)|x(k)=\xi_{k-}^i)}{\sum_{j=1}^{N} w_{k-}^i p(z(k)|x(k)=\xi_{k-})^i}$$

$$\text{and } \xi_k^i = \xi_{k-}^i \tag{11}$$

Resample for substitution of the degenerated particles.

Time update: compute state vector $x(k+1)$ according to

$$p(x(k+1)|Z) = \sum_{i=1}^{N} w_k^i \delta_{\xi_k^i}(x(k))$$
$$\text{where } \ \xi_k^i \sim p(x(k+1)|x(k) = \xi_k^i) \tag{12}$$

The stages of state vector estimation with the use of the particle filtering algorithm are depicted in Fig. 2.

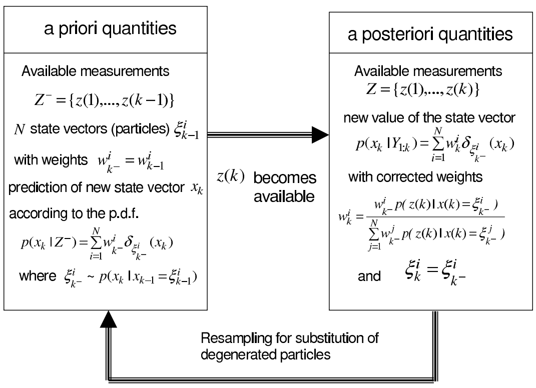


Fig. 2. Schematic diagram of the Particle Filter loop

B. Resampling issues in particle filtering

The algorithm of particle filtering which is described through Eq. (9) and Eq. (10) has a significant drawback: after a certain number of iterations k, almost all the weights w_k^i become 0. To avoid this, resampling is performed which substitutes the particles of low importance with those of higher importance. The particles $\{\xi_k^1, \cdots, x_k^N\}$ are chosen according to the probabilities $\{w_k^1, \cdots, w_k^N\}$. The resampling procedure of $(\xi_k^i, w_k^i \ i = 1, \cdots, N)$ is carried out through previous sorting in decreasing order of the particle weights. This will result into $w^{s[1]} > w^{s[2]} > \cdots > w^{s[N]}$. A random numbers generator is used and the resulting numbers $u^{i:N} \sim U[0,1]$ fall in the partitions of the interval $[0,1]$. The width of these partitions is w^i and thus a redistribution of the particles is generated. For instance, in a wide partition of width w^j will be assigned more particles than to a narrow partition of width w^m (see Fig. 3).

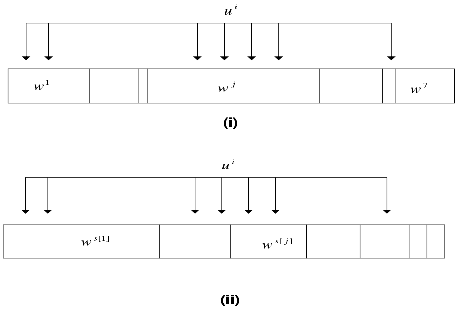


Fig. 3. Multinomial resampling: (i) conventional resampling, (ii) resampling with sorted weights

IV. Simulation Results

A. EKF-based Sensor Fusion for Vehicle Localization

Sensor fusion algorithms can be classified into three different groups: (i) fusion based on probabilistic models (e.g. Particle Filtering), (ii) fusion based on least squares techniques (e.g. Kalman Filtering), and (iii) intelligent fusion (e.g. fuzzy logic) [18]. This paper is concerned with cases (i) and (ii). The application of EKF to the fusion of data that come from different sensors is examined first [19]. A unicycle robot is considered. Its continuous-time kinematic equation is:

$$\dot{x}(t) = v(t)cos(\theta(t)), \ \dot{y}(t) = v(t)sin(\theta(t)), \ \dot{\theta}(t) = \omega(t) \tag{13}$$

which is a simplified model of a car-like robot studied in [20]. Encoders are placed on the driving wheels and provide a measure of the incremental angles over a sampling period T. These odometric sensors are used to obtain an estimation of the displacement and the angular velocity of the vehicle $v(t)$ and $\omega(t)$, respectively. These encoders introduce incremental errors, which result in an erroneous estimation of the orientation θ. To improve the accuracy of the vehicle's localization, measurements from sonars can be used. The distance measure of sonar i from a neighboring surface P_j is thus taken into account (see Fig. 4 and 5). Sonar measurements may be affected by white Gaussian noise and also by crosstalk interferences and multiples echoes.

The inertial coordinates system OXY is defined. Furthermore the coordinates system $O'X'Y'$ is considered (Fig. 4). $O'X'Y'$ results from OXY if it is rotated by an angle θ (Fig. 4). The coordinates of the center of the wheels axis with respect to OXY are (x, y), while the coordinates of the sonar i that is mounted on the vehicle, with respect to $O'X'Y'$ are $x_i^{'}, y_i^{'}$. The orientation of the sonar with respect to $OX'Y'$ is $\theta_i^{'}$. Thus the coordinates of the sonar with respect to OXY are (x_i, y_i) and its orientation is θ_i, and are given by:

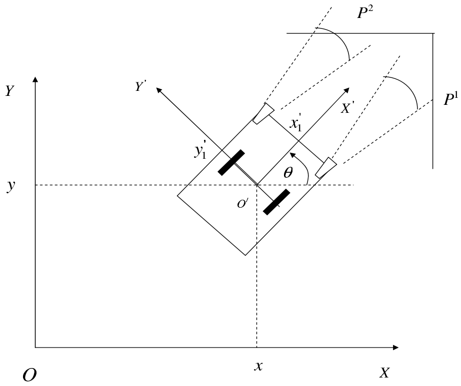


Fig. 4. Mobile robot with odometric sensors

$$\begin{aligned} x_i(k) &= x(k) + x_i^{'} sin(\theta(k)) + y_i^{'} cos(\theta(k)) \\ y_i(k) &= y(k) - x_i^{'} cos(\theta(k)) + y_i^{'} sin(\theta(k)) \\ \theta_i(k) &= \theta(k) + \theta_i \end{aligned} \tag{14}$$

Each plane P^j in the robot's environment can be represented by P_r^j and P_n^j (Fig. 5), where (i) P_r^j is the normal distance of the plane from the origin O, (ii) P_n^j is the angle between the normal line to the plane and the x-direction.

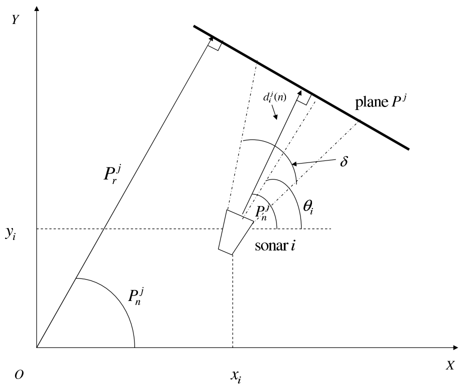


Fig. 5. Orientation of the sonar i

The sonar i is at position $x_i(k), y_i(k)$ with respect to the inertial coordinates system OXY and its orientation is $\theta_i(k)$. Using the above notation, the distance of the sonar i, from the plane P^j is represented by P_r^j, P_n^j (see Fig. 5):

$$d_i^j(k) = P_r^j - x_i(k) cos(P_n^j) - y_i(k) sin(P_n^j) \tag{15}$$

where $P_n^j \, \epsilon \, [\theta_i(n) - \delta/2, \theta_i(n) + \delta/2]$, and δ is the width of the sonar beam. Assuming a constant sampling period $\Delta t_k = T$ the measurement equation is $z(k+1) = \gamma(x(k)) + v(k)$, where $z(k)$ is the vector containing sonar and odometer measures and $v(k)$ is a white noise sequence $\sim N(0, R(kT))$. The dimension p_k of $z(k)$ depends on the number of sonar sensors. The measure vector $z(k)$ can be decomposed in two subvectors

$$\begin{aligned} z_1(k+1) &= [x(k) + v_1(k), y(k) + v_2(k), \theta(k) + v_3(k)] \\ z_2(k+1) &= [d_1^j(k) + v_4(k), \cdots, d_{n_s}^j(k) + v_{3+n_s}(k)] \end{aligned} \tag{16}$$

with $i = 1, 2, \cdots, n_s$, where n_s is the number of sonars, $d_i^j(k)$ is the distance measure with respect to the plane P^j provided by the i-th sonar and $j = 1, \cdots, n_p$ where n_p is the number of surfaces. By definition of the measurement vector one has that the output function $\gamma(x(k))$ is given by $\gamma(x(k)) = [x(k), y(k), \theta(k), d_1^1(k), d_2^2(k), \cdots, d_{n_s}^{n_p}]^T$. The robot state is $[x(k), y(k), \theta(k)]^T$ and the control input is denoted by $U(k) = [u(k), \omega(k)]^T$.

In the simulation tests, the number of sonar is taken to be $n_s = 1$, and the number of planes $n_p = 1$, thus the measurement vector becomes $\gamma(x(k)) = [x(k), y(k), \theta(k), d_1^1]^T$. To obtain the extended Kalman Filter (EKF), the kinematic model of the vehicle is linearized about the estimates $\hat{x}(k)$ and $\hat{x}^-(k)$ the control input $U(k-1)$ is applied.

The *measurement update* of the EKF is

$$\begin{aligned} &K(k) = P^-(k) J_\gamma^T(\hat{x}^-(k)) [J_\gamma(\hat{x}^-(k)) P^-(k) J_\gamma^T(\hat{x}^-(k)) + R(k)]^{-1} \\ &\hat{x}(k) = \hat{x}^-(k) + K(k)[z(k) - \gamma(\hat{x}^-(k))] \\ &P(k) = P^-(k) - K(k) J_\gamma^T P^-(k) \end{aligned}$$

The *time update* of the EKF is

$$\begin{aligned} &P^-(k+1) = J_\phi(\hat{x}(k)) P(k) J_\phi^T(\hat{x}(k)) + Q(k) \\ &\hat{x}^-(k+1) = \phi(\hat{x}(k)) + L(k) U(k) \end{aligned}$$

where $L(n) = \begin{pmatrix} Tcos(\theta(k)) & 0 \\ Tsin(\theta(k)) & 0 \\ 0 & T \end{pmatrix}$

and $J_\phi(\hat{x}(k)) = \begin{pmatrix} 1 & 0 & -v(k) sin(\theta) T \\ 0 & 1 & -v(k) cos(\theta) T \\ 0 & 0 & 1 \end{pmatrix}$,

while $Q(k) = diag[\sigma^2(k), \sigma^2(k), \sigma^2(k)]$, with $\sigma^2(k)$ chosen to be 10^{-3} and $\phi(\hat{x}(k)) = [\hat{x}(k), \hat{y}(k), \hat{\theta}(k)]^T$, $\gamma(\hat{x}(k)) = [, \hat{x}(k), \hat{y}(k), \hat{\theta}(k), d(k)]^T$, i.e.

$$\gamma(\hat{x}(k)) = \begin{pmatrix} \hat{x}(k) \\ \hat{y}(k) \\ \hat{\theta}(k) \\ P_r^j - x_i(k)) cos(P_n^j) - y_i(k) sin(P_n^j) \end{pmatrix} \tag{17}$$

Assuming one sonar $n_s = 1$, and one plane P^1, $n_p = 1$ in the mobile robot's neighborhood one gets $J_\gamma^T(\hat{x}^-(k)) = [J_{\gamma_1}(\hat{x}^-(k)), J_{\gamma_2}(\hat{x}^-(k)), J_{\gamma_3}(\hat{x}^-(k)), J_{\gamma_4}(\hat{x}^-(k))]^T$, i.e.

$$J_\gamma^T(\hat{x}^-(k)) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -cos(P_n^j) & -sin(P_n^j) & \{x_i^{'} cos(\theta - P_n^j) - y_i^{'} sin(\theta - P_n^j)\} \end{pmatrix} \tag{18}$$

The vehicle is steered by a dynamic feedback linearization control algorithm which is based on PD control [21]:

$$\begin{aligned} u_1 &= \ddot{x}_d + K_{p_1}(x_d - x) + K_{d_1}(\dot{x}_d - \dot{x}) \\ u_2 &= \ddot{y}_d + K_{p_2}(y_d - y) + K_{d_2}(\dot{y}_d - \dot{y}) \\ \dot{\xi} &= u_1 cos(\theta) + u_2 sin(\theta) \\ v &= \xi, \;\; \omega = \frac{u_2 cos(\theta) - u_1 sin(\theta)}{\xi} \end{aligned} \tag{19}$$

The following initialization is assumed (see Fig. 6):

- vehicle's initial position in OXY: $x(0) = 0m$, $y(0) = 0m$, $\theta(0) = 45.0^o$.
- position of the sonar in $O'X'Y'$: $x_1^{'} = 0.5m$, $y_1^{'} = 0.5m$, $\theta_1^{'} = 0^o$.
- position of the plane P^1: $P_r^1 = 15.5m$, $P_n^1 = 45^o$.
- state noise $w(k) = 0$, $\hat{P}(0) = diag[0.1, 0.1, 0.1]$, and $R = diag[10^{-3}, 10^{-3}, 10^{-3}, 10^{-3}]$.
- Kalman Gain $K(k) \,\epsilon\, R^{3\times 4}$.
- desirable trajectory: starts at $x_d(0) = 0, y_d(0) = 0$, and forms a 45^o with the OX axis.

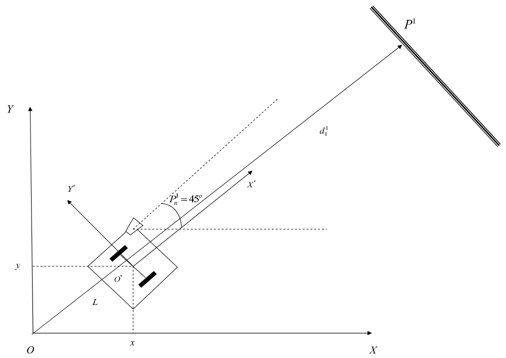


Fig. 6. Desirable trajectory of the autonomous vehicle i

The use of EKF for fusing the data that come from odometric and sonar sensors provides an estimation of the state vector $[x(t), y(t), \theta(t)]$ and enables the successful application of nonlinear steering control of Eq. (19). The obtained resutls are depicted in Fig. 7.

B. PF-based Sensor Fusion for Autonomous Vehicle Localization

The particle filter can also provide solution to the sensor fusion problem. The mobile robot model described in Eq. (13), and the control law given in Eq. (19) are used again. The number of particles was set to $N = 1000$.

The *measurement update* of the PF is $p(x(k)|Z) = \sum_{i=1}^{N} w_k^i \delta_{\xi_{k^-}^i}(x(k))$ with $w_k^i = \frac{w_{k^-}^i p(z(k)|x(k)=\xi_{k^-}^i)}{\sum_{j=1}^{N} w_k^j p(z(k)|x(k)=\xi_{k^-}^j)}$ where the measurement equation is given by $\hat{z}(k) = z(k) + v(k)$ with $z(k) = [x(k), y(k), \theta(k), d(k)]^T$, and $v(k)$ =measurement noise.

The *time update* of the PF is $p(x(k+1)|Z) = \sum_{i=1}^{N} w_k^i \delta_{\xi_k^i}(x(k))$ where $\xi_k^i \sim p(x(k+1)|x(k) = \xi_k^i)$ and the state equation is $\hat{x}^- = \phi(x(k)) + L(k)U(k)$, where $\phi(x(k))$, $L(k)$, and $U(k)$ are defined in subsection IV-A. At each run of the time update of the PF, the state vector estimation $\hat{x}^-(k+1)$ is calculated N times, starting each time from a different value of the state vector ξ_k^i. The obtained results are given in Fig. 8.

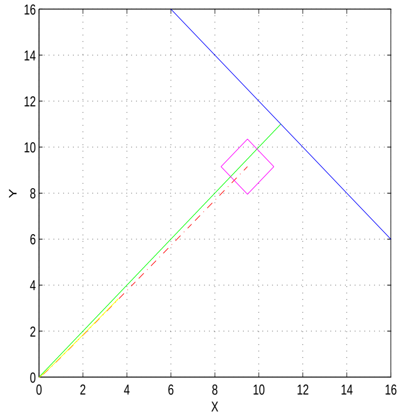


Fig. 7. Desirable trajectory (continuous line) and obtained trajectory using EKF fusion based on odometric and sonar measurements $(-.)$

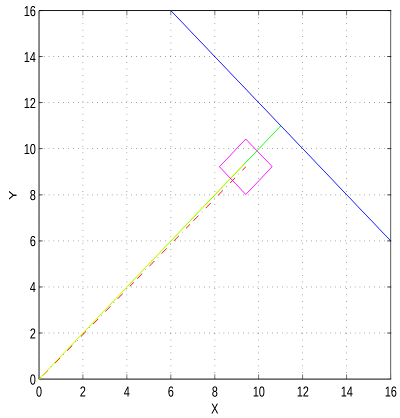
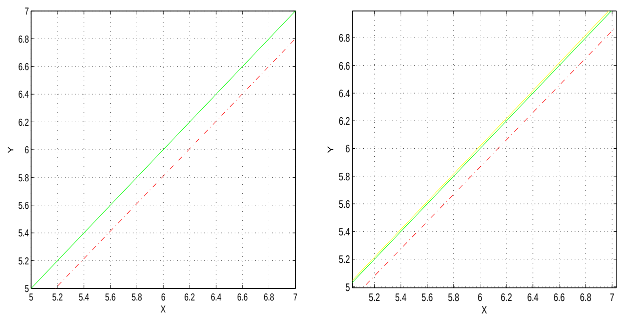


Fig. 8. Desirable trajectory (continuous line) and obtained trajectory using PF fusion based on odometric and sonar measurements $(-.)$

From the simulation experiments depicted it can be deduced that the particle filter has better performance than the EKF in the problem of estimation of the state vector of the mobile robot, without being subject in the constraint of Gaussian distribution for the obtained measurements (see fig. 9). The number of particles influences the performance of the particle filter algorithm. The accuracy of the estimation succeeded by the PF algorithm improves as the number of particles increases. The initialization of the particles, (state vector estimates) may also affect the convergence of the PF towards the real value of the state vector of the monitored system. It should be also noted that the calculation time is a critical parameter for the suitability of the PF algorithm for real-time applications. When it is necessary to use more particles, improved hardware and some new technologies, such as making parallel processing available to embedded systems, enable the PF to be implemented in real-time systems []8.



(a) **(b)**

Fig. 9. Precision of trajectory tracking (a) using EKF-based sensor fusion (−.) with respect to the desirable trajectory (continuous line) (b) using PF-based sensor fusion (−.) with respect to the desirable trajectory (continuous line)

V. Conclusions

Extended Kalman and Particle filtering have been tested in the problem of estimation of the state vector of a mobile robot through the fusion of position measurements coming from odometric and sonar sensors. The paper has summarized the basics of the Extended Kalman Filter, which is the most popular approach to implement sensor fusion in nonlinear systems. The EKF is a linearization technique, based of a first-order Taylor expansion of the nonlinear state functions and the nonlinear measurement functions of the state model. In the EKF, the state distribution is approximated by a Gaussian random variable. Although the EKF is a fast algorithm, the underlying series approximations can lead to poor representations of the nonlinear functions and the associated probability distributions. As a result, the EKF can sometimes be divergent.

To overcome these weekness of the EKF as well as the constraint of the Gaussian state distribution, particle filtering has been introduced. Whereas the EKF makes a Gaussian assumption to simplify the optimal recursive state estimation, the particle filter makes no assumptions on the forms of the state vector and measurement probability densities. In the particle filter a set of weighted particles (state vector estimates evolving in parallel) is used to approximate the posterior distribution of the state vector. An iteration of the particle filter includes particle update and weights update. To succeed the convergence of the algorithm at each iteration resampling takes place through which particles with low weights are substituted by particles of high weights.

Simulations have been carried out to give a comparison of the performance of the EKF and the particle filter algorithm in the problem of mobile robot localization, through the fusion of measurements coming from different sensors. These simulations have shown that the particle filter is superior than the EKF in terms of the accuracy of the state vector estimation. The performance of the particle filter algorithm depends on the number of particles and their initialization. It can be seen that the PF algorithms succeeds better estimates of the mobile robot's state vector as the number of particles increases, but on the expense of higher computational effort.

References

[1] E.W. Kamen and J.K. Su: Introduction to Optimal Estimation. *Springer*, 1999.

[2] G.G. Rigatos and S.G. Tzafestas, Extended Kalman Filtering for Fuzzy Modelling and Multi-Sensor Fusion, Mathematical and Computer Modelling of Dynamical Systems, *Taylor and Francis*, vol. 13, no.3, 2007.

[3] J. Fichou, F. LeGland and L. Mevel, Particle-Based Methods for Parameter Estimation and Tracking: Numerical Experiments, *Publication Interne IRISA*, No 1604, Rennes, France, 2004.

[4] B. Ristic, M.S. Arulampalam, and N.J. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications, *Artech House*, 2004.

[5] G.G. Rigatos, Particle Filtering for state estimation in nonlinear industrial systems, 2nd IC-EpsMsO 2007, 2nd International Conference on Experiments / Process / System Modelling / Simulation & Optimization, Athens, Greece, July 2007 .

[6] N.J. Gordon, D.J. Salmond and A.F.M. Smith, A novel approach to nonlinear / non-Gaussian Bayesian state estimation, *IEE Proceedings on Radar and Signal Processing*,vol. 140, pp. 107-113, 1993.

[7] K. Watanabe and S.G. Tzafestas, Learning algorithms for neural networks with the Kalman Filters. *J. Intelligent Robotic Systems*, vol. 3, pp. 305-319, 1990.

[8] N. Yang, W.F.Tian, Z.H. Jin, and C.B. Zhang, Particle Filter for sensor fusion in a land vehicle navigation system, Measurement Science and Technology, *Institute of Physics Publishing*, vol. 16, pp. 677-681, 2005.

[9] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics, *MIT Press*, 2005.

[10] F. Caron, M. Davy, E. Duflos and P. Vanheeghe, Particle Filtering for Multi-Sensor Data Fusion with Switching Observation Models: Applications to Land Vehicle Positioning, *IEEE Transactions on Signal Processing*, vol. 55, no.6, pp. 2703-2719, 2007.

[11] F. Gustafsson, F. Gunnarsson, N. Bergman and U. Forssel, Particle Filters for Positioning, Navigation and Tracking, *IEEE Transactions on Signal Processing - Special Issue on Monte Carlo Methods for Statistical Signal Processing*, vol. 50, no.2, pp. 425-437, 2002.

[12] P. Li and V. Kadirkamanathan, Particle Filtering Based Likelihood Ratio Approach to Fault Diagnosis in Nonlinear Stochastic Systems, *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, vol. 31, no.3, pp. 337-343, 2001.

[13] V. Kadirkamanathan, M.H. Jaward, S.G. Fabri and M. Kadirkamanathan, Particle Filters for Recursive Model Selection in Linear and Nonlinear System Identification, in Proc. 39th IEEE Conference on Decision and Control, vol. 3, pp. 2391-2396, 2000.

[14] R. Karlsson and M. Norrlöf, Bayesian Position Estimation of an Industrial Robot Using Multiple Sensors, in. Proc. 2004, *IEEE International Conference on Control Applications*, Taipei, Taiwan, pp. 303-308, 2004.

[15] G.W. Ng, Intelligent Systems - Fusion, Tracking and Control, *Research Studies Press Ltd*, 2003.

[16] F. Campillo, F. Cérou, F. LeGland and R. Rakotozafy, Particle and cell approximations for nonlinear filtering, *Rapport de Recherche INRIA*, No. 2567, June, 1995.

[17] D. Crisan and A. Doucet, A Survey of Convergence Results on Particle Filtering Methods for Practitioners, *IEEE Transactions on Signal Processing*, vol. 50, no3, pp.736-746,2002.

[18] R.C. Luo and M.G. Kay, Multisensor Integration and Fusion in Intelligent Systems. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 3, pp. 901-931, 1989.

[19] L. Jetto, S. Longhi and G. Venturini, Development and Experimental Validation of an Adaptive Extended Kalman Filter for the Localization of Mobile Robots. *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, 1999.

[20] G.G. Rigatos, Fuzzy Stochastic Automata for Intelligent Vehicle Control. *IEEE Transactions on Industrial Electronics*, vol. 50, no. 1, pp. 76-79, 2003.

[21] Oriolo G., De Luca A. and Vendittelli M.:, WMR Control Via Dynamic Feedback Linearization: Design, Implementation and Experimental Validation. *IEEE Transactions on Control Systems Technology*, vol. 10, no. 6, pp.835-852, 2002.