Parallel parking control of autonomous vehicles using Extended Kalman and Particle Filtering

Gerasimos G. Rigatos
Unit of Industrial Automation
Industrial Systems Institute
26504, Rion Patras, Greece
Email: grigat@isi.gr

***Abstract*—The paper studies implementation of parallel parking for autonomous vehicles using sensor fusion based on Extended Kalman Filtering and Particle Filtering. The vehicle is steered along a reference path while the vehicle's state vector is not directly available but is estimated by fusing measurements coming from distributed sensors. At a second stage the estimated state vector is used by a nonlinear controller (flatness-based controller) in-order to track the parking trajectory. It is shown that sensor fusion can provide reliable estimation of the vehicle's motion and can be used for successful completion of the parallel parking task and fault tolerant control.**

I. Introduction

The parallel parking task has attracted considerable attention in the area of mobile robotics since it has practical significance for the automotive industry [1-5]. A rectangular parking area is considered and a reference frame is defined so as its axes to coincide with the rectangle's side (see Fig. 1, Fig. 5 and Fig. 6). The vehicle starts from an initial position out of the rectangle and moves backwards. After performing appropriate maneuvers the vehicle should be found completely inside the parking area with its longitudinal axis to be aligned with the horizontal axis of the reference system [6]. Collisions with the boundaries of the parking area are not allowed, while several other constraints about the geometry of the vehicle's trajectory and the vehicle's velocity can be imposed. To perform the parallel parking task usually three stages can be distinguished: (i) detection of a suitable parking area, (ii) path planning, i.e. selection of the trajectory to be followed by the vehicle's center of wheels axis while moving towards the final position, (iii) control of the vehicle's motion. Parallel parking can be performed (a) in direct motion, i.e. the vehicle moves only backwards, along a single continuous path, until it reaches its final position (b) in recursive motion, i.e. the vehicle moves backwards and forwards and performs sequential maneuvers along different paths with different velocity sign, before reaching its goal point [1,6].

In this paper state estimation-based nonlinear control for direct-motion parallel parking will be examined. A robotic unicycle and a rectangular parking area are considered and a collision-free path that connects the initial to the final position of the vehicle is defined (see Fig. 5 and Fig. 6). The objective is to steer the vehicle along the reference path when the vehicle's state vector is not directly available but is estimated by fusing measurements coming from distributed sensors (such as odometric sensors, laser trackers and sonars or even cameras). At a second stage the estimated state vector is used by a nonlinear controller (flatness-based controller) in-order to make the mobile robot track the desired trajectory. Sensor fusion can provide reliable estimation of the vehicle's motion and can be used for fault tolerant control, since there is redundancy in the measurements used by the control loop [7-11].

The estimation of the robot's state's vector from position measurements can be carried out with the use of filtering algorithms, such as the Extended Kalman Filter and the Particle Filter [9-10]. Two different types of measurements will be considered: (i) measurements of the distance covered by the robot which are provided by the robot's encoders and (ii) measurements of the distance from a reference surface (blue line in Fig. 5 and Fig. 6) which can be provided by a sonar, laser tracker or camera. It is well known that the optimal filter for linear models with Gaussian noise is the Kalman Filter. 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 kind of approximation such as model linearisation in the Extended Kalman Filter (EKF). 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. 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 [9-11].

To overcome the EKF shortcomings, a new kind of nonlinear filtering method, the so-called Particle Filter (PF), has been proposed [12-13]. The Particle Filter is based on Monte-Carlo sampling from a state vectors distribution. Particle filtering has improved performance over the the EKF, since it can provide optimal estimation in nonlinear non-Gaussian state-space models. 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, resampling takes place at each iteration through which particles with low weights are substituted by particles of high weights [14-15]. Particle filters can estimate the system states sufficiently when the number of particles (estimations of the state vectors which evolve in parallel) is sufficiently large. However, in terms of computation time the Particle Filter is more demanding than the EKF.
The structure of the paper is as follows: In Section II 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. In Section IV simulation experiments are carried out to evaluate the performance of the Extended Kalman Filter and the Particle Filter in sensor fusion-based state estimation and control for automated parallel parking. Finally, in Section V concluding remarks are stated.

II. EKF-based Sensor Fusion for direct-motion parallel parking

The kinematic model of an autonomous vehicle (robotic unicycle) is considered (see Fig. 1). This is given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} v \\ \omega \end{pmatrix} \qquad (1)$$

where $v(t)$ is the speed of the vehicle, θ is the angle between the transversal axis of the vehicle and axis OX, and ω is the angular velocity. Moreover, it holds $\omega = \frac{v}{L}tan(\phi)$, where ϕ is the angle of the steering wheel with respect to the transversal axis of the vehicle, and L is the length of the vehicle (the sign of v and ϕ depends on whether the vehicle moves forward or backwards). Encoders are placed on the driving wheels and provide a measure of the incremental angles over a sampling period T. These odometric sensors cab be used to obtain an estimation of the speed and of the angular velocity of the vehicle $v(t)$ and $\omega(t)$, respectively. However, the encoders introduce incremental errors, which results 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. 1 and Fig. 2. Sonar measurements may be affected by noise (which is not always white Gaussian) and also by crosstalk interferences and multiples echoes.
The inertial coordinates system OXY is defined. Furthermore the coordinates system $O'X'Y'$ is considered (see Fig. 1. $O'X'Y'$ results from OXY if it is rotated by an angle θ. 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 $O'X'Y'$ 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 [7]:

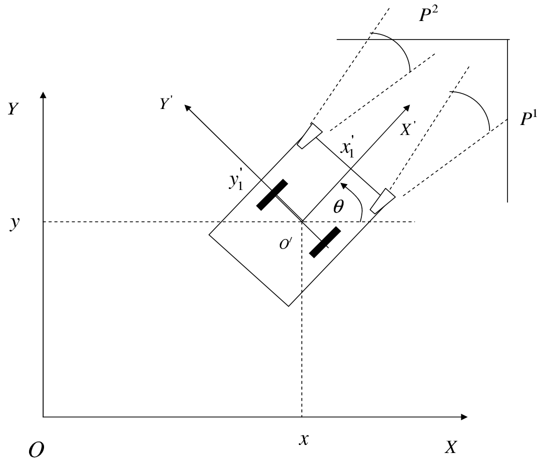

Fig. 1. Mobile robot with odometric sensors

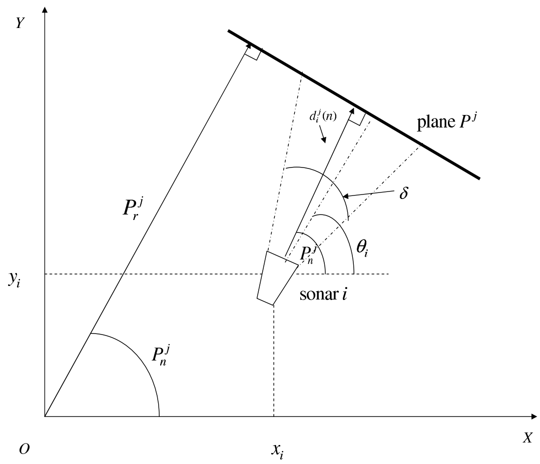

Fig. 2. Orientation of the i-th sonar

$$\begin{aligned} x_i(k) &= x(k) + x_i^{'}sin(\theta(k)) + y_i^{'}cos(\theta(k)) \\ y_i(k) &= y(k) - x_i^{'}cos(\theta(k)) + y_i^{'}sin(\theta(k)) \\ \theta_i(k) &= \theta(k) + \theta_i^{'} \end{aligned} \qquad (2)$$

Each plane (reference surface) P^j in the robot's environment can be represented by P_r^j and P_n^j (Fig. 2), 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. 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 sonar i, from plane P^j is represented by P_r^j, P_n^j (see Fig. 2):

$$d_i^j(k) = P_r^j - x_i(k)cos(P_n^j) - y_i(k)sin(P_n^j) \qquad (3)$$

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

$$\begin{array}{c} z_1(k+1) = [x(k)+v_1(k), y(k)+v_2(k), \theta(k)+v_3(k)] \\ z_2(k+1) = [d_1^j(k)+v_4(k), \cdots, d_{n_s}^j(k)+v_{3+n_s}(k)] \end{array} \tag{4}$$

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 mobile robot's state is $[x(k), y(k), \theta(k)]^T$ and the control input is denoted by $U(k) = [v(k), \omega(k)]^T$.

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 given in Eq. (1) is linearized about the estimates $\hat{x}(k)$ and $\hat{x}^-(k)$, and the control input $U(k)$ is applied. It is noted that $\hat{x}^-(k)$ is the estimation of the state vector before the k-th measurement to become available and $\hat{x}(k)$ is the estimation of the state vector after the k-th measurement has been obtained.

The *measurement update* of the EKF is

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

The *time update* of the EKF is

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

$$\text{where } L(k) = \begin{pmatrix} Tcos(\theta(k)) & 0 \\ Tsin(\theta(k)) & 0 \\ 0 & T \end{pmatrix} \tag{5}$$

and

$$\text{and } J_\phi(\hat{x}(k)) = \begin{pmatrix} 1 & 0 & -v(k)sin(\theta)T \\ 0 & 1 & -v(k)cos(\theta)T \\ 0 & 0 & 1 \end{pmatrix} \tag{6}$$

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.

$$\gamma(\hat{x}(k)) = \begin{pmatrix} \hat{x}(k) \\ \hat{y}(k) \\ \hat{\theta}(k) \\ P_r^j - x_i(k))cos(P_n^j) - y_i(k)sin(P_n^j) \end{pmatrix} \tag{7}$$

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.

$$J_\gamma^T(\hat{x}^-(k)) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -cos(P_n^j) & -sin(P_n^j) & x_i^{'}cos(\theta - P_n^j) - y_i^{'}sin(\theta - P_n^j) \end{pmatrix} \tag{8}$$

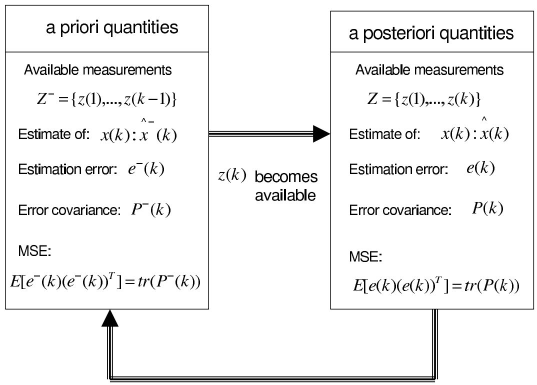

Fig. 3. Schematic diagram of the Extended Kalman Filter loop

The stages of EKF are depicted in Fig. 3. The vehicle is steered by a flatness-based controller. In the case of the autonomous vehicle of Eq. (1) the flat output is the cartesian position $\eta = (x, y)$ of the center of the wheels axis (see Fig. 1). The state variables of the vehicle's model as well as the control input can be expressed as functions of the flat output and its derivatives [4-5, 10].

$$\begin{array}{c} u_1 = \ddot{x}_d + K_{p_1}(x_d - x) + K_{d_1}(\dot{x}_d - \dot{x}) \\ u_2 = \ddot{y}_d + K_{p_2}(y_d - y) + K_{d_2}(\dot{y}_d - \dot{y}) \\ \dot{\xi} = u_1 cos(\theta) + u_2 sin(\theta) \\ v = \xi, \ \ \omega = \frac{u_2 cos(\theta) - u_1 sin(\theta)}{\xi} \end{array} \tag{9}$$

III. PF-based Sensor Fusion for direct-motion parallel parking

As in the case of the Extended Kalman Filter the Particle Filter consists of the measurement update (correction stage) and the time update (prediction stage) [12-13]

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

$$p(x(k-1)|Z^-) = \sum_{i=1}^{N} w_{k-1}^i \delta_{\xi_{k-1}^i}(x(k-1)) \tag{10}$$

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

$$\begin{array}{c} p(x(k)|Z^-) = \sum_{i=1}^{N} w_{k-1}^i \delta_{\xi_{k^-}^i}(x(k)) \\ \text{with } \ \xi_{k^-}^i \sim p(x(k)|x(k-1) = \xi_{k-1}^i) \end{array} \tag{11}$$

The meaning of Eq. (11) 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 $x(k-1) = \xi_{k-1}^i$. 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 is performed using Eq. (11). 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

$$p(x(k)|Z) = \sum_{i=1}^{N} w_k^i \delta_{\xi_{k^-}^i}(x(k))$$
$$\text{where } w_k^i = \frac{w_{k^-}^i p(z(k)|x(k)=\xi_{k^-}^i)}{\sum_{j=1}^{N} w_{k^-}^j p(z(k)|x(k)=\xi_{k^-}^j)} \tag{12}$$

Eq. (12) denotes the corrected value for the state vector. According to above, the recursion of the Particle Filter can be formulated 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

$$p(x(k)|Z) = \sum_{i=1}^{N} w_k^i \delta_{\xi_{k^-}^i}(x(k))$$
$$\text{with corrected weights } \ w_k^i = \frac{w_{k^-}^i p(z(k)|x(k)=\xi_{k^-}^i)}{\sum_{j=1}^{N} w_{k^-}^i p(z(k)|x(k)=\xi_{k^-}^i)}$$
$$\text{and } \xi_k^i = \xi_{k^-}^i \tag{13}$$

Finally, *resampling* is performed for substitution of the degenerated particles (substitution of particles of low importance with those of higher importance).

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

$$p(x(k+1)|Z) = \sum_{i=1}^{N} w_k^i \delta_{\xi_k^i}(x(k))$$
$$\text{where } \ \xi_k^i \sim p(x(k+1)|x(k) = \xi_k^i) \tag{14}$$

The stages of state vector estimation with the use of the Particle Filter algorithm are depicted in Fig. 4.

IV. Simulation tests

The autonomous vehicle described in Eq. (1), and the control law given in Eq. (9) were considered. The use of EKF for fusing the data that come from odometric and sonar sensors provided an estimation of the state vector $[x(t), y(t), \theta(t)]$ and enabled the successful application of nonlinear steering control of Eq. (9). The obtained results are depicted in Fig. 5. The vehicle was steered along a reference path which assured avoidance of collisions with obstacles. Direct parallel parking

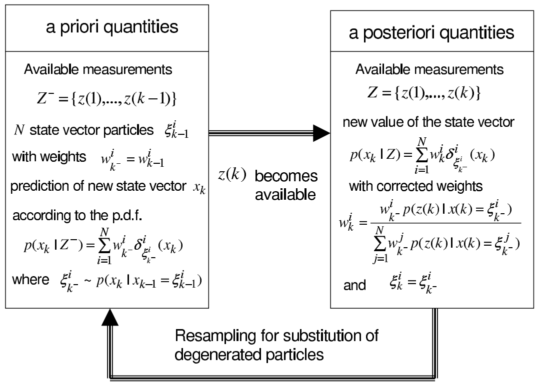

Fig. 4. Schematic diagram of the Particle Filter loop

was assumed to be completed when the center of the wheels axis reached the desirable position and the vehicle is aligned to the OX axis.
In the case of Particle Filtering the number of particles was set to $N = 1000$. At each run of the time update of the PF, the state vector estimation $\hat{x}^-(k+1)$ was calculated N times, starting each time from a different value of the state vector ξ_k^i. The obtained results are given in Fig. 6.

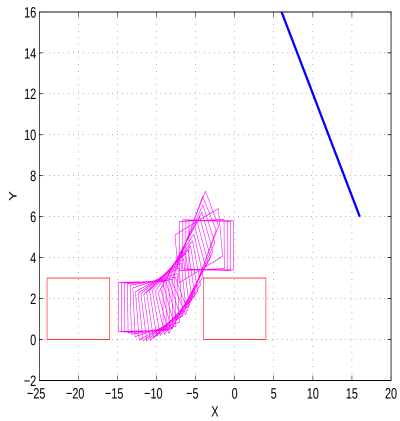

Fig. 5. Maneuvers performed by the autonomous vehicle for the completion of parallel parking when the vehicle's state vector is estimated using Extended Kalman Filtering

From the simulation experiments it can be observed that both the EKF and Particle Filter have good performance in the problem of estimation of the state vector of the autonomous vehicle. Both approaches enable sensor fusion-based control and implementation of direct-motion parallel parking. However, the Particle Filter is not subject to the constraint of Gaussian distribution for the obtained measurements. 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

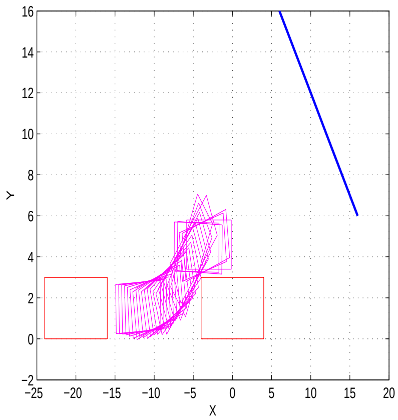

Fig. 6. Maneuvers performed by the autonomous vehicle for the completion of parallel parking when the vehicle's state vector is estimated using Particle Filtering

suitability of the PF algorithm for real-time applications. When it is necessary to use more particles, avoidance of shorting in the resampling procedure, improved hardware and parallel processing a available to embedded systems, enable the PF to be implemented in real-time systems [9-10,17-18].

V. Conclusions

The paper has given results on the implementation of parallel parking for autonomous vehicles using sensor fusion based on Extended Kalman Filtering and Particle Filtering. The vehicle's state vector has been estimated by fusing measurements coming from distributed sensors and a nonlinear control law (flatness-based control) has been employed to steer the vehicle along a reference path, thus making possible to carry out direct-motion parallel parking. A first conclusion is that sensor fusion can provide reliable estimation of the vehicle's motion and can be used for fault tolerant control of autonomous vehicles, since there is redundancy in the measurements processed by the control loop.

A second conclusion is about the performance of sensor fusion-based state estimation using Extended Kalman Filtering and Particle Filtering. It has been noted that although the EKF is a fast algorithm, the underlying series expansion can lead to poor representations of the nonlinear functions and of the associated probability distributions. As a result, the EKF can sometimes be divergent. Additionally, the EKF algorithm assumption that the state distribution is approximated by a Gaussian random variable, is not always valid. On the other hand, it has been noted that the Particle Filter makes no assumptions about the forms of the state vector and the output measurement p.d.f (probability density functions). Additionally, in contrast to EKF in the Particle Filter there in no need for analytical calculation of Jacobians.

Simulation tests have shown that the performance of the particle filter algorithm depends on the number of particles and their initialization. It has been observed that the PF algorithm succeeds better estimates of the autonomous vehicle state vector as the number of particles increases, but at the same time it requires more computational power. This research work can be extended towards sensor fusion-based control and autonomous navigation of several types of AGVs (automatic guided vehicles) or UAVs (unmanned aerial vehicles).

References

[1] F. Gomez-Bravo, F. Cuesta and A. Ollero, Parallel and diagonal parking in nonholonomic autonomous vehicles, Engineering Applications of Artificial Intelligence, Elsevier, vol.14, pp. 419-434, 2001.

[2] J. Hermosillo and S. Sekhavat, Feedback Control of a Bi-steerable Car Using Flatness Application to Trajectory Tracking, In Proc 2003, IEEE 2003 American Control Conference, Denver Colorado, pp. 3567-3572, 2003.

[3] D. Vissière, D.E. Chang, and N. Petit, Experiments of trajectory generation and obstacle avoidance for a UGV, In Proc. ACC 2007, IEEE 2007 American Control Conference, pp. 2828-2835, New York, July 2007.

[4] P. Brault, H. Mounier, N. Petit and P. Rouchon, Flatness based tracking control of a manoeuvrable vehicle : the π-Car, In Proc. MTNS00, Mathematical Theory of Networks and Systems, Perpignan, France, 2000.

[5] J. Villagra, B. d'Andrea-Novel, H. Mounier, and M. Pengov, Flatness-based vehicle steering control strategy with SDRE feedback gains tuned via a sensitivity approach, IEEE Transactions on Control Systems Technology, vol. 15, no. 3, pp. 554-565, 2007.

[6] G.G. Rigatos, S.G. Tzafestas, and G.A. Evangelidis, Reactive parking control of a nonholonomic vehicle via a Fuzzy Learning Automaton, IEE Proceedings : Control Theory and Applications, vol. 148, pp. 169-179, 2001.

[7] G. Oriolo, A. De Luca, M. Vendittelli, WMR Control Via Dynamic Feedback Linearization: Design, Implementation and Experimental Validation, IEEE Transactions on Control Systems Technology, vol. 10, pp. 835-852, 2002.

[8] G.G. Rigatos, S.G. Tzafestas, Extended Kalman Filtering for Fuzzy Modelling and Multi-Sensor Fusion, Mathematical and Computer Modelling of Dynamical Systems, Talor and Francis, vol. 13, pp. 251-266, 2007.

[9] G.G. Rigatos, Autonomous robots navigation using flatness-based control and multi-sensor fusion, in: Robotics, Automation and Control, I-Tech Education and Publishing KG, Vienna Austria, 2008.

[10] G.G. Rigatos, Particle Filtering for state estimation in industrial robotic systems, IMeche Journal of Systems and Control Engineering, vol. 222, pp. 437-455, 2008 .

[11] E.W. Kamen and J.K. Su, Introduction to optimal estimation, Springer, 2009

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

[13] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, MIT Press, 2005.

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

[15] Q. Zhang, F. Campillo, F. Cérou and F. Legland, Nonlinear fault detection and isolation based on bootstrap particle filters, Proc. of the 44th IEEE Conference on Decision and Control, Seville Spain, Dec. 2005

[16] J.P. Laumont, P.E. Jacobs, M. Taix and M. Murray, A motion planner for nonholonomic mobile robots, IEEE Transactions on Robotics and Automation, vol. 10, no. 5, pp. 577-593, 1994.

[17] M. Bolic, P.M. Djuric and H. Hong, Resampling algorithms and architectures for distributed Particle Filters, IEEE Transactions on Signal Processing, vol. 53, no.7, pp. 2242-2450, 2005.

[18] 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.