A DENSITY-ASSISTED PARTICLE FILTER FOR MOBILE ROBOT LOCALIZATION WITH UNCERTAIN ENVIRONMENT MAP

Paulo R. A. Silva and Marcelo G. S. Bruno

Instituto Tecnológico de Aeronáutica
São José dos Campos SP 12228-900, Brazil
ph:(55-12) 3947-6906; email: {pras,bruno}@ita.br

ABSTRACT

We present in this paper a modified density-assisted particle filter for indoor mobile robot localization in a situation where the environment map is subject to random uncertainties and is not perfectly known to the tracker. The proposed filter jointly estimates the robot's pose and the environment map parameters combining raw measurements from a range-finding laser scanner and the robot's odometric data. Experiments with real data show promising results even in adverse scenarios with abrupt maneuvers and heavily cluttered environments.

Index Terms— Monte Carlo methods, Nonlinear estimation, State-space methods.

1. INTRODUCTION

We introduced in [1] an improved sequential Monte Carlo (SMC) filter [2] for tracking the pose of a mobile robot using a parametric model of the environment. The algorithm in [1] assumed however perfect knowledge of the environment map, which is not realistic in most practical scenarios. In this paper, we modify the filter to account for possible uncertainties in the environment model parameters and test the modified algorithm using real data.

We model the robot's environment as in [3] by a set of straight lines specified by their respective start and end points. Unlike in [1], however, we assume that the parameters that characterize each line are independent realizations of fixed (time-invariant) random variables with a uniform prior distribution. The proposed modified SMC filter jointly tracks then the dynamic (time-varying) robot's pose and the line parameters, assimilating features that are extracted from raw data generated by a laser scanner mounted on the robot. As in [1], a preliminary clutter suppression algorithm is applied to the raw data before the features of interest may be extracted.

Joint estimation of dynamic state variables and static parameters remains an open and challenging problem in the SMC literature. As pointed out in [4], the conventional solution of extending the state vector to include the unknown parameters is bound to fail with static parameters due to extreme particle degeneracy. Other solutions based on assuming an artificial random walk model for the unknown parameters, see e.g. [5], require on the other hand a suitable specification of the variance of the random drift, which may be in turn difficult to select. In this paper, we resort to an alternative approach known as density-assisted particle filtering [6], which is based on using the existing particle set to build a parametric approximation of the joint posterior distribution of the dynamic state variables and the fixed unknown parameters and then resampling according to that approximation at the next time step.

The paper is divided into 5 sections. Section 1 is this Introduction. In Section 2, we present the theory behind density-assisted particle filtering in a more general framework than previously described in [6]. In Section 3, we review the state and observation models from [1] for the robot pose tracking problem and show how the ideas in Section 2 may be used to design a tracking filter assuming uncertain environment map parameters. The algorithm proposed in Section 3 is then tested with real data collected by a Magellan Pro Robot moving in a heavily cluttered room. Experimental results are shown and discussed in Section 4. Finally, we present in Section 5 the conclusions of our work.

2. DENSITY-ASSISTED PARTICLE FILTERS FOR JOINT STATE AND PARAMETER ESTIMATION

Let $\{\mathbf{x}_k\}$, $k \geq 0$, and $\{\mathbf{y}_k\}$, $k \geq 1$, be two sequences of random vectors specified by the dynamic model

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \underline{\theta}) + \mathbf{u}_k \quad (1)$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \underline{\theta}) + \mathbf{v}_k \quad (2)$$

where $\{\mathbf{u}_k\}$, $k \geq 0$, and $\{\mathbf{v}_k\}$, $k \geq 1$, are two mutually independent, identically distributed (i.i.d) random sequences; $\underline{\theta}$ is an unknown, time-invariant random parameter vector independent of $\mathbf{x}_0$, $\{\mathbf{u}_k\}$, and $\{\mathbf{v}_k\}$; and $\mathbf{f}_k$ and $\mathbf{h}_k$ are two (generally nonlinear) known functions.

Given an observed sequence $\{\mathbf{y}_k\}$, $k \geq 1$, our goal is to derive a recursive algorithm for the computation of $E\,\{\mathbf{g}(\mathbf{x}_k, \underline{\theta})$

The work of the first author was supported by CAPES, Brazil.

$\mid \mathbf{y}_{1:k}\}$ where E stands for the expected value (or expectation), $\mathbf{y}_{1:k} = \left[\mathbf{y}_1^T \ldots \mathbf{y}_k^T\right]^T$, and $\mathbf{g}$ is any arbitrary (measurable) function of the hidden state vector $\mathbf{x}_k$ and of the unknown parameter vector $\underline{\theta}$. Assuming that $\mathbf{x}_k$ and $\underline{\theta}$ are (absolutely) continuous random vectors with an associated joint probability density function (pdf) denoted by the lowercase letter p, we note first that

$$E\left\{\mathbf{g}(\mathbf{x}_k, \underline{\Theta}) \mid \mathbf{y}_{1:k}\right\} = \int\int \mathbf{g}(\mathbf{x}_k, \underline{\theta})\, p(\mathbf{x}_k, \underline{\theta} \mid \mathbf{y}_{1:k}) d\underline{\theta}\, d\mathbf{x}_k\,. \tag{3}$$

The integral on the right-hand side of (3) is expanded in turn as

$$\int\int \left[\mathbf{g}(\mathbf{x}_k, \underline{\theta})\, \frac{p(\mathbf{y}_k \mid \mathbf{x}_k, \underline{\theta})\, p(\mathbf{x}_k \mid \underline{\theta}, \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1})} \times \quad p(\underline{\theta} \mid \mathbf{y}_{1:k-1})\right] d\underline{\theta}\, d\mathbf{x}_k. \tag{4}$$

Recalling next that $p(\mathbf{x}_k \mid \underline{\theta}, \mathbf{y}_{1:k-1})$ can be computed as $\int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \underline{\theta})\, p(\mathbf{x}_{k-1} \mid \underline{\theta}, \mathbf{y}_{1:k-1})\, d\mathbf{x}_{k-1}$, we write the desired expected value as

$$\frac{1}{p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1})} \int\int\int \left[\mathbf{g}(\mathbf{x}_k, \underline{\theta})\, p(\mathbf{y}_k \mid \mathbf{x}_k, \underline{\theta})\, p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \underline{\theta})\, p(\mathbf{x}_{k-1} \mid \underline{\theta}, \mathbf{y}_{1:k-1})\, p(\underline{\theta} \mid \mathbf{y}_{1:k-1})\right] d\underline{\theta}\, d\mathbf{x}_{k-1}\, d\mathbf{x}_k. \tag{5}$$

In the sequel, let $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \underline{\theta}, \mathbf{y}_k) > 0$ be an arbitrary proposal pdf and define

$$w(\mathbf{x}_k, \underline{\theta}, \mathbf{x}_{k-1}) = \frac{p(\mathbf{y}_k \mid \mathbf{x}_k, \underline{\theta})\, p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \underline{\theta})}{q(\mathbf{x}_k \mid \mathbf{x}_{k-1},\, \underline{\theta},\, \mathbf{y}_k)}. \tag{6}$$

We re-write then the integral in (5) as

$$\frac{1}{p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1})} \int\int\int \left[\mathbf{g}(\mathbf{x}_k, \underline{\theta}) w(\mathbf{x}_k, \underline{\theta}, \mathbf{x}_{k-1}) q(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \underline{\theta}, \mathbf{y}_k)\, p(\mathbf{x}_{k-1} \mid \underline{\theta}, \mathbf{y}_{1:k-1})\, p(\underline{\theta} \mid \mathbf{y}_{1:k-1})\right] d\underline{\theta}\, d\mathbf{x}_{k-1}\, d\mathbf{x}_k. \tag{7}$$

The proportionality constant $C_k = 1/p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1})$ can be eliminated in turn by dividing (7) by $C_k \int\int\int w(\mathbf{x}_k, \underline{\theta}, \mathbf{x}_{k-1})$ $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \underline{\theta}, \mathbf{y}_k)$ $p(\mathbf{x}_{k-1} \mid \underline{\theta}, \mathbf{y}_{1:k-1})\, p(\underline{\theta} \mid \mathbf{y}_{1:k-1}) d\underline{\theta}$ $d\mathbf{x}_{k-1}$ $d\mathbf{x}_k$, which is equal to one. Finally, a Monte Carlo estimate of $E\left\{\mathbf{g}(\mathbf{X}_k, \underline{\Theta}) \mid \mathbf{y}_{1:k}\right\}$ can be obtained by drawing N_p samples

- $\underline{\theta}^{(j)} \sim p(\theta \mid \mathbf{y}_{1:k-1})$
- $\mathbf{x}_{k-1}^{(j)} \sim p(\mathbf{x}_{k-1} \mid \underline{\theta}^{(j)}, \mathbf{y}_{1:k-1})$,
- $\mathbf{x}_k^{(j)} \sim q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(j)}, \underline{\theta}^{(j)}, \mathbf{y}_k)$,

and making

$$E\left\{\mathbf{g}(\mathbf{x}_k, \underline{\Theta}) \mid \mathbf{y}_{1:k}\right\} \approx \sum_{j=1}^{N_p} w_k^{(j)} \mathbf{g}(\mathbf{x}_k^{(j)}, \underline{\theta}^{(j)}) \tag{8}$$

where

$$w_k^{(j)} = \frac{w(\mathbf{x}_k^{(j)}, \underline{\theta}^{(j)}, \mathbf{x}_{k-1}^{(j)})}{\sum_{i=1}^{N_p} w(\mathbf{x}_k^{(i)}, \underline{\theta}^{(i)}, \mathbf{x}_{k-1}^{(i)})}$$

with $w(\mathbf{x}_k, \underline{\theta}, \mathbf{x}_{k-1})$ defined as in (6). It can be shown that the approximation (8) is asymptotically unbiased and converges almost surely to the desired expectation as $N_p \to \infty$.

2.1. Density-Assisted Bootstrap Particle Filter (DAPF)

Unfortunately, it is normally impossible to sample directly from $p(\underline{\theta} \mid \mathbf{y}_{1:k-1})$ and $p(\mathbf{x}_{k-1} \mid \underline{\theta}, \mathbf{y}_{1:k-1})$ as described in Section 2. An alternative approach known as *density-assisted particle filtering* was introduced originally in a less general framework in [6] and is based on using the sample set at instant k to build parametric approximations of $p(\underline{\theta} \mid \mathbf{y}_{1:k})$ and $p(\mathbf{x}_k \mid \underline{\theta}, \mathbf{y}_{1:k})$ that can be then easily resampled from at the next time step $k+1$. A recursive bootstrap algorithm to accomplish that task is described in the following steps:

1. Set $k = 1$; make $\widehat{p}(\underline{\theta} \mid \mathbf{y}_{1:k-1}) = p(\underline{\theta})$ and $\widehat{p}(\mathbf{x}_{k-1} \mid \underline{\theta}, \mathbf{y}_{1:k-1}) = p(\mathbf{x}_0)$.

2. For $j = 1, \ldots, N_p$

- Draw $\underline{\hat{\theta}}^{(j)} \sim \widehat{p}(\underline{\theta} \mid \mathbf{y}_{1:k-1})$.
- Draw $\mathbf{x}_{k-1}^{(j)} \sim \widehat{p}(\mathbf{x}_{k-1} \mid \underline{\hat{\theta}}^{(j)}, \mathbf{y}_{1:k-1})$.
- Draw $\tilde{\mathbf{x}}_k^{(j)} \sim q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(j)}, \underline{\hat{\theta}}^{(j)}, \mathbf{y}_k)$.
- Compute the weights $w(\tilde{\mathbf{x}}_k^{(j)}, \hat{\theta}^{(j)}, \mathbf{x}_{k-1}^{(j)})$ according to (6).

3. Normalize the weights, and compute the estimates

$$\hat{\mathbf{z}}_{k|k} = \sum_{j=1}^{N_p} w_k^{(j)} \left[(\tilde{\mathbf{x}}_k^{(j)})^T\, (\underline{\hat{\theta}}^{(j)})^T\right]^T$$

$$\hat{\mathbf{P}}_{k|k} = \sum_{j=1}^{N_p} w_k^{(j)} \left[\mathbf{z}_k^{(j)} - \hat{\mathbf{z}}_{k|k}\right] \left[\mathbf{z}_k^{(j)} - \hat{\mathbf{z}}_{k|k}\right]^T$$

where $\mathbf{z}_k^{(j)} = \left[(\tilde{\mathbf{x}}_k^{(j)})^T (\underline{\hat{\theta}}^{(j)})^T\right]^T$.

4. Build new parametric approximations $\widehat{p}(\underline{\theta} \mid \mathbf{y}_{1:k})$ and $\widehat{p}(\mathbf{x}_k \mid \underline{\theta}, \mathbf{y}_{1:k})$ matching the sample means and covariance matrices found in step 3.

5. Make $k = k + 1$ and go back to step 2.

3. APPLICATION EXAMPLE: MOBILE ROBOTICS

Let $\mathbf{x}_k = [x_k\, y_k\, \gamma_k]^T$ be an unknown random vector that collects, at instant k, the robot's pose (x_k, y_k) and orientation angle, γ_k, with respect to a fixed (non-inertial) coordinate system, henceforth referred to as the environment system. The robot's state $\mathbf{x}_{k+1}$ at instant $k+1$ is obtained by the nonlinear stochastic model [1, 3]

$$\underbrace{\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \gamma_{k+1} \end{bmatrix}}_{\mathbf{x_{k+1}}} = \underbrace{\begin{bmatrix} x_k + d_k cos(\frac{\Delta\gamma_k}{2} + \gamma_k) \\ y_k + d_k sin(\frac{\Delta\gamma_k}{2} + \gamma_k) \\ \gamma_k + \Delta\gamma_k \end{bmatrix}}_{\mathbf{f}(\mathbf{x}_k, \underline{\varepsilon}_k)} + \underbrace{\begin{bmatrix} u_k^{(x)} \\ u_k^{(y)} \\ u_k^{(\gamma)} \end{bmatrix}}_{\mathbf{u}_k} \tag{9}$$

where $\underline{\varepsilon}_k = (d_k, \Delta\gamma_k)$ is obtained from odometric data collected by the robot and, for the purposes of our discussion, is deterministic and known for each instant k. The random sequence $\{\mathbf{u}_k\}$ is assumed on the other hand to be i.i.d and Gaussian with $\mathbf{u}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$.

3.1. Observation Model

Assume that, for a rectangular room, there are N walls in the field of view of the robot's sensor at instant k, where $N = 1, \ldots, 4$. For $i = 1, \ldots, N$, the extracted features used for data assimilation at instant k are [1, 3] the perpendicular distances, $\rho_{k,i}$, from the robot's centroid to each detected wall, and the respective orientation angles, $\alpha_{k,i}$, of each detected wall line, measured with respect to the moving (inertial) coordinate system of the robot, henceforth referred to as the *robot system*. Next, let ρ_i^m, $i = 1, \ldots, N$, denote the perpendicular distances from the origin of the *environment* coordinate system to each detected wall, and let α_i^m, $i = 1, \ldots, N$, denote the corresponding orientation angles of each wall line with respect to the same fixed coordinate system. The feature vector $\mathbf{y}_{k,i} = [\rho_{k,i}\ \alpha_{k,i}]^T$ for the ith detected wall at instant k is given by the nonlinear model (modified from [3])

$$\mathbf{y}_{k,i} = \underbrace{\begin{bmatrix} \zeta_{k,i}(\rho_i^m - \sqrt{x_k^2 + y_k^2} cos(\alpha_i^m - \beta_k)) \\ \alpha_i^m - \gamma_k + \xi_{k,i} \end{bmatrix}}_{\mathbf{h}_{k,i}(\mathbf{x}_k, \underline{\theta})} + \mathbf{v}_{k,i} \quad (10)$$

where $\beta_k = \tan^{-1}(y_k/x_k)$, $\{\mathbf{v}_{k,i}\}$, $k \geq 1$, is a sequence of Gaussian random vectors with zero mean and covariance matrices $\mathbf{R}_{k,i}$, and the constants $\zeta_{k,i}$ and $\xi_{k,i}$ are technical correction coefficients which ensure that $\rho_{k,i} > 0$ and $\alpha_{k,i} \in [-\pi, \pi]$ for all relative positions between the robot and the walls. In real-data applications, a preliminary clutter suppression algorithm is used to eliminate data points that come from unwanted objects in the room. A feature extraction routine combining a weighted Hough transform and least-squares line fitting is used then to compute $\rho_{k,i}$ and $\alpha_{k,i}$ for each detected line in the field of view of the robot, see [1, 3] for further details.

3.2. DAPF Pose Tracking

Contrary to our previous work [1], we assume in this paper that the environment map is parameterized by an unknown, fixed (time-invariant) realization of a random vector $\underline{\theta} = [\theta_1\ \theta_2\ \ldots \theta_p]^T$, where each parameter θ_l is assumed independent of θ_j for $l \neq j$ and distributed a priori according to a uniform pdf with support in the (known) range $\left[\theta_{l,\min},\ \theta_{l,\max}\right]$. Given a sequence of observed features $\mathbf{y}_{1:k}$, we recursively compute a Monte Carlo approximation of the expected robot pose $E[\mathbf{x}_k \mid \mathbf{y}_{1:k}]$ at instant k using the modified density-assisted bootstrap particle filter of Section 2.1.

Importance Function Approximation In the implementation of the DAPF, we approximate the optimal importance function [2], $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(j)}, \underline{\theta}^{(j)}, \mathbf{y}_k) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(j)}, \underline{\theta}^{(j)}, \mathbf{y}_k)$ by linearizing the observation model (10) around $\mathbf{f}(\mathbf{x}_{k-1}^{(j)}, \underline{\varepsilon}_{k-1})$ and then making $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(j)}, \underline{\theta}^{(j)}, \mathbf{y}_k) \approx \mathcal{N}(\mathbf{m}_k^{(j)}, \mathbf{\Sigma}_k^{(j)})$ where

$$\mathbf{\Sigma}_k^{(j)} = \left[\mathbf{Q}^{-1} + (\mathbf{H}_k^{(j)})^T \mathbf{R}_k^{-1} (\mathbf{H}_k^{(j)})\right]^{-1} \quad (11)$$

$$\begin{aligned} \mathbf{m}_k^{(j)} &= (\Sigma_k^{(j)}) \Big\{ \mathbf{Q}^{-1}\mathbf{f}(\mathbf{x}_{k-1}^{(j)}, \underline{\varepsilon}_{k-1}) \\ &+ (\mathbf{H}_k^{(j)})^T \mathbf{R}_k^{-1} \Big[\mathbf{y}_k - \mathbf{h}_k(\mathbf{f}(\mathbf{x}_{k-1}^{(j)}, \underline{\varepsilon}_{k-1}), \underline{\theta}^{(j)}) \\ &+ \mathbf{H}_k^{(j)} \mathbf{f}(\mathbf{x}_{k-1}^{(j)}, \underline{\varepsilon}_{k-1}) \Big] \Big\} . \end{aligned} \quad (12)$$

In (12), $\mathbf{y}_k$ is a column vector of variable dimension that collects the extracted features $\mathbf{y}_{k,i}$ corresponding to all walls that are in the field of view of the robot at instant k. Similarly, $\mathbf{R}_k$ is a block-diagonal matrix collecting the corresponding feature extraction covariance matrices $\mathbf{R}_{k,i}$ and $\mathbf{H}_k$ is a long matrix obtained by stacking the matrices $\mathbf{H}_{k,i}^{(j)} = \frac{\partial \mathbf{h}_{k,i}(\mathbf{x}_k, \underline{\theta}^{(j)})}{\partial \mathbf{x}_k}$ evaluated at $\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}^{(j)}, \underline{\varepsilon}_{k-1})$, for each detected wall i.

Parametric Approximation of the Posterior Density
In step 4 of the algorithm in Section 2.1, we use the following parametric approximations:
1. $\widehat{p}(\underline{\theta} \mid \mathbf{y}_{1:k}) = \prod_{i=1}^{p} \mathcal{B}(\theta_i; \lambda_{i,1}, \lambda_{i,2})$, where $\mathcal{B}(\theta_i; \lambda_{i,1}, \lambda_{i,2})$ is a shifted Beta pdf with (time-varying) shape parameters $\lambda_{i,1}$ and $\lambda_{i,2}$ matching the posterior sample means and sample variances for the parameter θ_i computed as in step 3 described in Section 2.1.
3. $\widehat{p}(\mathbf{x}_k \mid \hat{\underline{\theta}}^{(j)}, \mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{a}_k^{(j)}, \mathbf{P}_k^{(j)})$ where $\mathbf{a}_k^{(j)}$ and $\mathbf{P}_k^{(j)}$ are computed using the linear least-squares estimates

$$\mathbf{a}_k^{(j)} = \hat{\mathbf{x}}_{k,k} + \mathbf{P}_{k|k}^{\mathbf{x}_k,\theta} (\mathbf{P}_k^{\theta})^{-1} (\hat{\underline{\theta}}^{(j)} - \hat{\underline{\theta}}_k) \quad (13)$$

$$\mathbf{P}_k^{(j)} = \mathbf{P}_{k|k}^{\mathbf{x}} - \mathbf{P}_{k|k}^{\mathbf{x}_k,\theta} (\mathbf{P}_k^{\theta})^{-1} (\mathbf{P}_{k|k}^{\mathbf{x}_k,\theta})^T, \quad (14)$$

where $\hat{\mathbf{x}}_{k,k}$, $\hat{\theta}_k$, $\mathbf{P}_{k|k}^{\mathbf{x}}$, $\mathbf{P}_{k|k}^{\mathbf{x}_k,\theta}$, and $\mathbf{P}_k^{\theta}$ are obtained from the sample mean $\widehat{\mathbf{z}}_{k|k}$ and the sample covariance matrix $\hat{\mathbf{P}}_{k|k}$ from Section 2.1.

Feature Extraction
The features $\mathbf{y}_k$ must be computed from the raw laser data at each iteration of the algorithm. The first step in the feature extraction process is to convert the sensor measurements from the robot coordinate system to the environment coordinate system using a predicted pose $\hat{\mathbf{x}}_{k|k-1}$ based on past observations only. In a sequential Monte Carlo framework, the predicted pose can be approximated by sampling a set of auxiliary particles $\overline{\mathbf{x}}_k^{(j)} \sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(j)}, \hat{\underline{\theta}}^{(j)})$ and then making $\hat{\mathbf{x}}_{k|k-1} \approx (1/N_p) \sum_j \overline{\mathbf{x}}_k^{(j)}$. The procedure is somewhat

simplified in the particular application in this paper since the kinematic model in equation (9) does not depend on the unknown parameter vector $\underline{\theta}$. Following the coordinate conversion, the algorithm decides which lines fall within the field of view of the sensor and, given a set of validated measurements within a specified range around the walls, extracts the features $\mathbf{y}_k$ using a weighted Hough transform combined with a linear least-squares line estimation procedure as described in [1, 3]. After the feature extraction routine is complete, the auxiliary particles $\overline{\mathbf{x}}_k^{(j)}$ are discarded and new samples $\tilde{\mathbf{x}}_k^{(j)}$ are drawn from the measurement-driven importance function $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(j)}, \underline{\theta}^{(j)}, \mathbf{y}_k)$.

4. EXPERIMENTAL RESULTS

We tested the proposed algorithm with *real data* recorded in a heavily cluttered room using a laser scanner mounted on a Magellan Pro robot. The room is represented by a set of four straight lines corresponding to each of its walls. The start and end points of each of the four straight lines are modeled as unknown uniform random variables defined on a range equal to the true parameter value plus or minus a 10 % uncertainty. In the implementation of the tracking filter, we assume for simplicity time-invariant, empirically-estimated covariance matrices $\mathbf{Q}$ and $\mathbf{R}_{k,i} = \tilde{\mathbf{R}}$, $i = 1, \ldots, 4$, given by $\tilde{\mathbf{R}} = \text{diag}(50^2, (\pi/5)^2)$, $Q(1,1) = 16.49$, $Q(2,2) = 5.24$, $Q(3,3) = 0.0000989$, $Q(1,2) = Q(2,1) = -4.18$, $Q(1,3) = Q(3,1) = 0.0229$, and $Q(2,3) = Q(3,2) = -0.0014$. The initial pose (x_0, y_0) is assumed Gaussian with standard deviation equal to 20 cm in both dimensions.

Figure 1 shows the robot trajectory estimated by the proposed DAPF tracker over 150 consecutive time steps. The filtered trajectory is superimposed to the trajectory obtained by the deterministic integration of the robot's odometric data (without any laser data fusion) and is compared to our best estimate of the ground truth. As expected, the trajectory predicted deterministically by the odometric data alone deviates over time from the real robot trajectory highlighting the need for data assimilation and stochastic modeling of the dynamic evolution of the robot's pose. We see from Figure 1 that the proposed DAPF tracker was capable however of tracking the robot's position fairly accurately even though the chosen trajectory includes abrupt turns close to corners where the robot's sensor has a very narrow field of view, leading to poor feature extraction.

5. CONCLUSIONS

We presented in this paper a modified density-assisted particle filter for dynamic pose tracking in mobile robotics assuming a scenario where the robot's environment map is subject to random uncertainty and is not perfectly known to the tracking filter. The proposed filter combines a kinematic model based on the integration of the robot's odometry and an observation model based on geometric features that are extracted from raw laser scan data. Experiments with real data show good results even in difficult scenarios where the robot makes abrupt maneuvers in regions where the field of view of the laser sensor is very narrow.

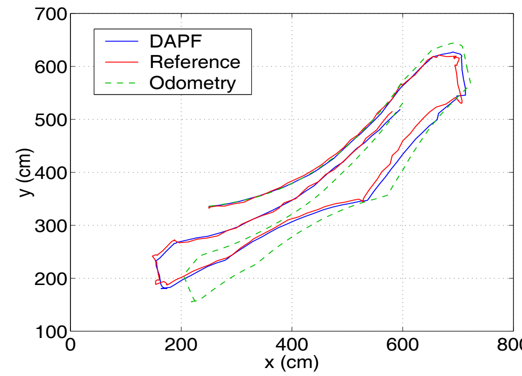

Fig. 1. Estimated robot trajectory using real data.

6. REFERENCES

[1] P. R. A. Silva and M. G. S. Bruno, "Mobile Robot Localization using Improved SIR Filters and Parametric Models of the Environment", *Proc. ICASSP 2006*, vol.III, pp. 17-20, Toulose, France, May 14-19, 2006.

[2] A. Doucet, S. J. Godsill, and C. Andrieu,"On sequential Monte Carlo sampling methods for Bayesian filtering," *Stat. Comput.*, vol.10, pp. 197-208, 2000.

[3] P. Jensfelt and H. I. Christensen, "Pose tracking using laser scanning and minimalistic environment models," *IEEE Trans. Robot. and Autom.*, v.17, n.2, pp. 138-147, April 2001.

[4] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadić, "Particle Methods for Change Detection, System Identification, and Control," *Proceedings of the IEEE*, vol.92, n.3, pp. 423-438, March 2004.

[5] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. Freitas, and N. J. Gordon, Editors, pp.197-223, New York: Springer-Verlag, 2001.

[6] P. M. Djurić, M. F. Bugallo, and J. Míguez, "Density assisted particle filters for state and parameter estimation," *Proc. ICASSP 2004*, vol.II, pp. 701-704, Montreal, Canada, May 17-21, 2004.