ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

C. Manes, A. Martinelli, F. Martinelli, P. Palumbo

MOBILE ROBOT LOCALIZATION BASED ON A POLYNOMIAL APPROACH

R. 635 Gennaio 2006

Costanzo Manes – Dipartimento di Ingegneria Elettrica, Università degli studi di L'Aquila, Monteluco di Roio, 67040 L'Aquila, Italy. E-mail: manes@ing.univaq.it.

Agostino Martinelli – Autonomous System Laboratory, ETHZ, Zurich, Switzerland and with Inria Rhône Alpes, St Ismier Cedex, France. E-mail: agostino.martinelli@ieee.org.

Francesco Martinelli – Dipartimento di Informatica, Sistemi e Produzione, Università di Roma "Tor Vergata", via del Politecnico, I-00133, Rome, Italy. E-mail: martinelli@disp.uniroma2.it.

Pasquale Palumbo – Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30, 00185 Roma, Italy. Email: palumbo@iasi.rm.cnr.it.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica, CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.rm.cnr.it
URL: http://www.iasi.rm.cnr.it

Abstract

This paper introduces a new analytical algorithm to perform the localization of a mobile robot using odometry and laser readings. Based on a polynomial approach, the proposed algorithm provides the optimal affine filter in a class of suitably defined estimators. The performance of the algorithm has been evaluated through simulations. The comparison with the standard Extended Kalman Filter shows that the proposed filter provides good estimates also in critical situations where the system nonlinearities cause a bad behavior of the EKF.

Key words: Mobile robot localization, Nonlinear filtering, Polynomial estimate

1. Introduction

In most cases, autonomous mobile robots are required to know precisely their position and orientation in order to successfully perform their mission. This is usually achieved by fusing proprioceptive data (gathered by sensors monitoring the motion of the vehicle, like encoders) with exteroceptive data (e.g. [2, 7, 8, 15]). One of the most common methods adopted to perform this fusion is the Extended Kalman Filter (EKF, e.g. see [8]).

Apart from very few cases, both the dynamics of a mobile robot and the link between the data gathered by the robot sensors and the robot configuration are nonlinear functions. As a result, the EKF is not an optimal filter. It introduces an approximation by linearizing these functions around the current estimated state. In many cases, this approximation can generate a divergence. The worst situation arises when the localization error is not bounded and in particular it monotonically grows during the navigation. This happens when the robot configuration is not observable. A typical example of this situation is the navigation in an unknown environment when the robot is equipped with sensors able to provide only the relative configuration of the robot with respect to other objects in the environment (e.g., range sensors or cameras). In this scenario, in order to perform the localization, the robot has to estimate simultaneously the global configuration of these objects and store them in a common absolute map. In other words, it has to solve the SLAM (Simultaneous Localization and Mapping) problem. It is well known that, in this context, the EKF diverges if the environment is large enough. A deep analysis of this divergence is carried out in [6].

In several situations, the robot configuration is observable. For instance, an a priori map is provided and/or the robot has absolute localization capabilities (e.g. is equipped with a GPS). Even in these cases, the linearization introduced by the EKF can be a too rough approximation leading therefore to the divergence. This can happen when the sensor data are delivered at a very low frequency with respect to the robot speed and/or the data are not precise enough.

In order to avoid the problems resulting from the system nonlinearities, usually numerical methods to approximate the posterior density function for the state are adopted. Many numerical approaches to the localization problem are based on the *Markov Localization* (e.g.[3], [10], [12], [17], [18]).

Another very successful numerical approach in this framework is the *Monte Carlo Localization* in [19], which is based on the particle filters [9, 13, 16].

In contrast with these numerical approaches, in this paper we develop an analytical method able to deal with the system nonlinearities. The methodology here adopted is based on a polynomial approach, which provided a great deal of results in the last decade in the framework of filtering linear [4], bilinear [5] or nonlinear [11] systems, both in the Gaussian or non-Gaussian case. The first step in the filter construction is to pre-process the measurements available from the robot in order to write the new output equations as a second-order polynomial transformation of a suitably defined extended state. Then a generation model of the pre-processed data is achieved in the form of a bilinear system (linear drift and multiplicative noise), to which the optimal linear filter is finally applied [5]. It is important to stress that the filter construction is not based on the linear approximation of the system but on the exact system equations. The computations in the paper are quite messy but the idea behind them is rather simple: introduce a transformation in the description of the system in order to obtain a form for which the optimal linear filter can be applied.

The simulations reported show the effectiveness of the proposed approach under several parameter settings, and the improvements w.r.t. the standard EKF.

The paper is organized as follows. In the next Section the case of a mobile robot equipped with encoder and range sensors, moving in a $2D$ environment is presented. We assume to know a map of the environment. The output transformation is described in Section III, together with its bilinear generation model. The filtering algorithm is provided in Section 4; simulation results are reported in Section V. Conclusions follow.

2. System notation and problem formulation

Consider the motion equations of a mobile robot moving in a 2D environment. Its configuration is described through the Cartesian coordinates x_t, y_t and the orientation angle θ_t. We assume that the dynamics verify the unicycle differential equations. A numerical integration of these equations provides:

$$x_{t+1} = x_t + \delta\rho_t \cos(\theta_t), \tag{2.1}$$
$$y_{t+1} = y_t + \delta\rho_t \sin(\theta_t), \tag{2.2}$$
$$\theta_{t+1} = \theta_t + \delta\theta_t, \tag{2.3}$$

where $t = 0, 1, \ldots$; $\delta\rho_t$ and $\delta\theta_t$ are the robot shift and rotation occurring during the sample time. According to the hypothesis of synchronous drive, both $\delta\rho_t$, $\delta\theta_t$ are given by the encoder sensors. Adopting a synchronous odometry error model (see e.g. [14]), where $\delta\rho_t^e$, $\delta\theta_t^e$ denote the encoder readings, we can model $\{\delta\rho_t\}$, $\{\delta\theta_t\}$ as independent sequences of independent Gaussian random variables, such that:

$$\delta\rho_t \sim \mathcal{N}\big(\delta\rho_t^e, K_\rho \delta\rho_t^e\big), \qquad \delta\theta_t \sim \mathcal{N}\big(\delta\theta_t^e, K_\theta \delta\rho_t^e\big), \tag{2.4}$$

where K_ρ and K_θ are positive constants characterizing the robot odometry. In practice, equation (2.4) assumes that the odometry is perfectly calibrated (mean value equal to the reading) and the variances increase linearly with the traveled distance (as in the diffusion motion). By introducing the following quantities:

$$\nu_{\rho,t} = \frac{\delta\rho_t - \delta\rho_t^e}{\sqrt{\delta\rho_t^e}}, \qquad \nu_{\theta,t} = \frac{\delta\theta_t - \delta\theta_t^e}{\sqrt{\delta\rho_t^e}}, \tag{2.5}$$

it comes that:

$$\delta\rho_t = \delta\rho_t^e + \sqrt{\delta\rho_t^e}\nu_{\rho,t}, \quad \text{with} \quad \nu_{\rho,t} = \mathcal{N}\big(0, K_\rho\big), \tag{2.6}$$
$$\delta\theta_t = \delta\theta_t^e + \sqrt{\delta\rho_t^e}\nu_{\theta,t}, \quad \text{with} \quad \nu_{\theta,t} = \mathcal{N}\big(0, K_\theta\big), \tag{2.7}$$

that is, the equations of system (2.1-2.3) become:

$$x_{t+1} = x_t + \delta\rho_t^e \cos(\theta_t) + \sqrt{\delta\rho_t^e}\nu_{\rho,t}\cos(\theta_t), \tag{2.8}$$
$$y_{t+1} = y_t + \delta\rho_t^e \sin(\theta_t) + \sqrt{\delta\rho_t^e}\nu_{\rho,t}\sin(\theta_t), \tag{2.9}$$
$$\theta_{t+1} = \theta_t + \delta\theta_t^e + \sqrt{\delta\rho_t^e}\nu_{\theta,t}. \tag{2.10}$$

As a matter of fact, also $\{\nu_{\rho,t}\}$, $\{\nu_{\theta,t}\}$ are independent sequences of independent Gaussian random variables.

The environment where the robot moves is perfectly known and can be represented by line segments. Furthermore, our robot is equipped with a laser range finder able to provide the

distance at m directions, each direction with angle θ_i w.r.t. the robot orientation θ_t. The equation of the environment line segment sensed in the laser direction i ($i = 1, 2, \dots, m$) at time t will be denoted by $y = m_{it}x + q_{it}$ (notice that we are assuming in the following, without loss of generality, $m_{it} < \infty$: the extension to the general case is straightforward) with $\rho_{i,t}$ being the corresponding laser range finder reading. We have:

$$\rho_{i,t} = \sqrt{(x_t - x_{p_{it}})^2 + (y_t - y_{p_{it}})^2} + n_{i,t}, \tag{2.11}$$

where $(x_{p_{it}}, y_{p_{it}})$ is the intersection between the line $y = m_{it}x + q_{it}$ describing the face of the object sensed by the i^{th} laser beam and the line $y = y_t + \tan(\theta_t + \theta_i)(x - x_t)$ characterizing the direction of observation of the same laser beam. $\{n_{i,t}\}$ are independent sequences of zero-mean independent random Gaussian variables, with variance $\zeta_{i,2}$ (in general we use the notation $\zeta_{i,j} = I\!E\big[n_{i,t}^j\big]$ to indicate the higher order moments) and, moreover, they are also independent of $\{\nu_{\rho,t}\}$, $\{\nu_{\theta,t}\}$.

The aim of the paper is to estimate both the position (x_t, y_t) and the orientation θ_t starting from the odometry and the readings $\rho_{i,t}$, $i = 1, \dots, m$.

3. A bilinear model for the robot

The first step in the filter construction is to pre-process the measurements available from the robot in order to write a new output equation, as a polynomial transformation of a suitably defined extended state.

3.1. The output equations

Consider the measurement equations (2.11) and substitute the values of the pair $(x_{p_{it}}, y_{p_{it}})$. Then, after computations:

$$\rho_{i,t} - n_{i,t} = \frac{|y_t - m_{it}x_t - q_{it}|}{d}, \tag{3.1}$$

where

$$d = \left|\big(\sin(\theta_i) - m_{it}\cos(\theta_i)\big)\cos(\theta_t) + \big(\cos(\theta_i) + m_{it}\sin(\theta_i)\big)\sin(\theta_t)\right|. \tag{3.2}$$

Let $c_t = \cos(\theta_t)$ and $s_t = \sin(\theta_t)$. For our purposes, it is convenient to define the vector $X(t) = (x_t, y_t, c_t, s_t)^T$, to take the squares of both terms in (3.1) and put it in the following form

$$0 = C_{i,0}(t) + C_{i,1}(t)X(t) + C_{i,2}(t)X^{[2]}(t) + D_i(t)X^{[2]}(t)N_{o,i}(t), \tag{3.3}$$

where $N_{o,i}(t) = n_{i,t}^2 - \zeta_{i,2} - 2\rho_{i,t}n_{i,t}$ and the matrices $C_{i,0}$, $C_{i,1}$, $C_{i,2}$, D_i are:

$$\begin{aligned}
C_{i,2}(t) &= \big[-m_{it}^2 \;\; m_{it} \;\; 0\;0 \;\; m_{it} \;\; -1\;0\;0\;0\;0 \; C_{i,2}(t)\big|_{11} \; C_{i,2}(t)\big|_{12} \; 0\;0 \; C_{i,2}(t)\big|_{15} \; C_{i,2}(t)\big|_{16}\big],\\
C_{i,1}(t) &= \big[-2m_{it}q_{it} \;\; 2q_{it} \;\; 0 \;\; 0\big],\\
C_{i,0}(t) &= -q_{it}^2,\\
D_i(t) &= \big[O_{1\times 10} \;\; D_i(t)\big|_{11} \;\; D_i(t)\big|_{12} \;\; 0 \;\; 0 \;\; D_i(t)\big|_{15} \;\; D_i(t)\big|_{16}\big],
\end{aligned} \tag{3.4}$$

with:

$$\begin{aligned}
&C_{i,2}(t)\big|_{11} = \big(\rho_{i,t}^2 + \zeta_{i,2}\big)\big(\sin(\theta_i) - m_{it}\cos(\theta_i)\big)^2,\\
&C_{i,2}(t)\big|_{12} = C_{i,2}(t)\big|_{15} = \big(\rho_{i,t}^2 + \zeta_{i,2}\big)\big(\sin(\theta_i) - m_{it}\cos(\theta_i)\big)\big(\cos(\theta_i) + m_{it}\sin(\theta_i)\big),\\
&C_{i,2}(t)\big|_{16} = \big(\rho_{i,t}^2 + \zeta_{i,2}\big)\big(\cos(\theta_i) + m_{it}\sin(\theta_i)\big)^2,\\
&D_i(t)\big|_{11} = \big(\sin(\theta_i) - m_{it}\cos(\theta_i)\big)^2,\\
&D_i(t)\big|_{12} = D_i(t)\big|_{15} = \big(\sin(\theta_i) - m_{it}\cos(\theta_i)\big)\big(\cos(\theta_i) + m_{it}\sin(\theta_i)\big),\\
&D_i(t)\big|_{16} = \big(\cos(\theta_i) + m_{it}\sin(\theta_i)\big)^2.
\end{aligned} \tag{3.5}$$

The superscript in square brackets in (3.3) denotes the Kronecker power of a vector (for a quick survey on the Kronecker product and its main properties, see the appendix in [5] and references therein). By taking into account the further constraint of the trigonometric identity:

$$\cos^2(\theta_t) + \sin^2(\theta_t) = 1 \quad \Longleftrightarrow \quad X_3^2(t) + X_4^2(t) = 1, \tag{3.6}$$

and defining $Y(t) = \big(Y_0(t), Y_1(t), \dots, Y_m(t)\big)^T$, with $Y_0(t) \equiv 1$, $Y_i(t) \equiv 0$, $i = 1, \dots, m$, equations (3.3) and (3.6) may be written in the following more compact form:

$$Y(t) = C_1(t)X(t) + C_2(t)X^{[2]}(t) + C_0(t) + H(t)X^{[2]}(t), \tag{3.7}$$

with $X(t) \in I\!R^n$, $n = 4$, $Y(t) \in I\!R^{m+1}$ and:

$$C_0(t) = \begin{bmatrix} 0 \\ C_{1,0}(t) \\ \vdots \\ C_{m,0}(t) \end{bmatrix}, \quad C_1(t) = \begin{bmatrix} O_{1\times n} \\ C_{1,1}(t) \\ \vdots \\ C_{m,1}(t) \end{bmatrix}, \quad C_2(t) = \begin{bmatrix} C_{0,2} \\ C_{1,2}(t) \\ \vdots \\ C_{m,2}(t) \end{bmatrix}, \quad H(t) = \begin{bmatrix} O_{1\times n^2} \\ D_1(t)N_{o,1}(t) \\ \vdots \\ D_m(t)N_{o,m}(t) \end{bmatrix}, \tag{3.8}$$

with $C_{0,2} = \begin{bmatrix} O_{1\times 10} & 1 & O_{1\times 4} & 1 \end{bmatrix}$ and $O_{r\times c}$ is an r by c zero matrix. Notice that $H(t)$ is a stochastic matrix.

3.2. The extended state dynamics

In order to obtain a generation model for the output equation (3.7), in this subsection the recursive equations of $X(t)$, which will be referred to as the *extended state* in the sequel, are achieved. According to (2.8-2.10), the extended state dynamics obeys the following bilinear law (i.e. linear drift and multiplicative noise):

$$X_1(t+1) = X_1(t) + \delta\rho_t^e X_3(t) + \sqrt{\delta\rho_t^e}\nu_{\rho,t}X_3(t), \tag{3.9}$$

$$X_2(t+1) = X_2(t) + \delta\rho_t^e X_4(t) + \sqrt{\delta\rho_t^e}\nu_{\rho,t}X_4(t), \tag{3.10}$$

with:

$$\begin{aligned}
X_3(t+1) = &\big(X_3(t)\cos(\delta\theta_t^e) - X_4(t)\sin(\delta\theta_t^e)\big)\cos\big(\sqrt{\delta\rho_t^e}\nu_{\theta,t}\big)\\
&-\big(X_4(t)\cos(\delta\theta_t^e) + X_3(t)\sin(\delta\theta_t^e)\big)\sin\big(\sqrt{\delta\rho_t^e}\nu_{\theta,t}\big),
\end{aligned} \tag{3.11}$$

and:

$$\begin{aligned} X_4(t+1) = \big(X_4(t)\cos(\delta\theta_t^e) + X_3(t)\sin(\delta\theta_t^e)\big)\cos\big(\sqrt{\delta\rho_t^e}\nu_{\theta,t}\big) \\ +\big(X_3(t)\cos(\delta\theta_t^e) - X_4(t)\sin(\delta\theta_t^e)\big)\sin\big(\sqrt{\delta\rho_t^e}\nu_{\theta,t}\big). \end{aligned} \tag{3.12}$$

Note that $\cos\big(\sqrt{\delta\rho_t^e}\nu_{\theta,t}\big)$ and $\sin\big(\sqrt{\delta\rho_t^e}\nu_{\theta,t}\big)$ are both non-Gaussian random variables with the following mean values:

$$\bar{\xi}_{c1,t} := I\!E\big[\cos\big(\sqrt{\delta\rho_t^e}\nu_{\theta,t}\big)\big] = e^{-\frac{\delta\rho_t^e K_\theta}{2}}, \tag{3.13}$$

$$I\!E\big[\sin\big(\sqrt{\delta\rho_t^e}\nu_{\theta,t}\big)\big] = 0. \tag{3.14}$$

In order to cope with zero-mean noises, denote $N_c(t) = \cos\big(\sqrt{\delta\rho_t^e}\nu_{\theta,t}\big) - \bar{\xi}_{c1,t}$ and $N_s(t) = \sin\big(\sqrt{\delta\rho_t^e}\nu_{\theta,t}\big)$. Then, equations (3.11-3.12) become:

$$\begin{aligned} X_3(t+1) = \big(X_3(t)\cos(\delta\theta_t^e) - X_4(t)\sin(\delta\theta_t^e)\big)\bar{\xi}_{c1,t} + \big(X_3(t)\cos(\delta\theta_t^e) - X_4(t)\sin(\delta\theta_t^e)\big)N_c(t) \\ -\big(X_4(t)\cos(\delta\theta_t^e) + X_3(t)\sin(\delta\theta_t^e)\big)N_s(t), \end{aligned} \tag{3.15}$$

$$\begin{aligned} X_4(t+1) = \big(X_4(t)\cos(\delta\theta_t^e) + X_3(t)\sin(\delta\theta_t^e)\big)\bar{\xi}_{c1,t} + \big(X_4(t)\cos(\delta\theta_t^e) + X_3(t)\sin(\delta\theta_t^e)\big)N_c(t) \\ +\big(X_3(t)\cos(\delta\theta_t^e) - X_4(t)\sin(\delta\theta_t^e)\big)N_s(t). \end{aligned} \tag{3.16}$$

The moments of $N_c(t)$ and $N_s(t)$ will be denoted by $\xi_{ci,t} = I\!E\big[N_c^i(t)\big]$, $\xi_{si,t} = I\!E\big[N_s^i(t)\big]$, $\xi_{ci,sj,t} = I\!E\big[N_c^i(t)N_s^j(t)\big]$. As it will be clearer in the sequel, they will be required to be finite and available up to order 4. According to their construction, it readily comes that $\xi_{c1,t} = \xi_{s1,t} = \xi_{s3,t} = \xi_{ci,sj,t} = 0$, $\forall\, t \geq 0$, $\forall\, i = 1,\ldots,4, \forall\, j = 1,3$. The explicit computation of the other moments is reported in the Appendix.

In summary, by using the further position $N(t) = \sqrt{\delta\rho_t^e}\nu_{\rho,t}$ we have, from (3.9), (3.10), (3.15) and (3.16):

$$X(t+1) = A(t)X(t) + S_1(t)X(t), \tag{3.17}$$

with $S_1(t) = BN(t) + B_1(t)N_c(t) + B_2(t)N_s(t)$ and:

$$A(t) = \begin{bmatrix} I_2 & \delta\rho_t^e\, I_2 \\ O_2 & \bar{\xi}_{c1,t}\, R(\delta\theta_t^e) \end{bmatrix}, \tag{3.18}$$

$$B = \begin{bmatrix} O_2 & I_2 \\ O_2 & O_2 \end{bmatrix}, \quad B_1(t) = \begin{bmatrix} O_2 & O_2 \\ O_2 & R(\delta\theta_t^e) \end{bmatrix}, \quad B_2(t) = \begin{bmatrix} O_2 & O_2 \\ O_2 & R(\delta\theta_t^e + \frac{\pi}{2}) \end{bmatrix}, \tag{3.19}$$

where O_k is a square matrix of 0's of order k, I_k is the identity matrix of order k and :

$$R(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{3.20}$$

is a rotation matrix of angle ϕ. Note that the first term in (3.17) is a linear drift; the other is a multiplicative noise. In the sequel, the moments of the Gaussian variable N will be indicated by $\eta_{i,t} = I\!E\big[N^i(t)\big]$. They will be useful up to degree 4 (recall that $\eta_{1,t} \equiv 0$, $\eta_{3,t} \equiv 0$).

3.3. A bilinear generation model

The output equation (3.7) can be seen both as a quadratic transformation of the extended state, and as a linear transformation of an *augmented state* $\mathcal{X}(t) \in \mathbb{R}^{n+n^2}$, whose components are the first and second order Kronecker powers of the extended state $\mathcal{X}(t) = \begin{pmatrix}\mathcal{X}_1^T(t) & \mathcal{X}_2^T(t)\end{pmatrix}^T$, $\mathcal{X}_1(t) = X(t)$, $\mathcal{X}_2(t) = X^{[2]}(t)$. Then:

$$Y(t) = \mathcal{C}(t)\mathcal{X}(t) + \mathcal{V}(t) + \mathcal{N}_o(t), \tag{3.21}$$

with $\mathcal{C}(t) = \begin{bmatrix}C_1(t) & C_2(t)\end{bmatrix}$; $\mathcal{V}(t) = C_0(t)$ is a deterministic drift, and $\mathcal{N}_o(t) = \mathcal{H}(t)\mathcal{X}(t) = H(t)\mathcal{X}_2(t)$ is a multiplicative noise.

In order to obtain a bilinear (w.r.t. $\mathcal{X}$) generation model of the output equation in (3.21), the Kronecker square of the extended state $X(t)$ is needed. The computation requires the binomial expansion of a Kronecker power, which is below reported for the ease of the reader for the second-order case (see [5] for more details):

$$(a+b)^{[2]} = a^{[2]} + b^{[2]} + M_1^2(n)(a \otimes b), \quad a, b \in \mathbb{R}^n, \tag{3.22}$$

with $\otimes$ denoting the Kronecker product among matrices and:

$$M_1^2(n) = I_n + \mathcal{C}_n; \tag{3.23}$$

$\mathcal{C}_n$ is the commutation matrix in $\{0,1\}^{n^2 \times n^2}$ according to which the identity $b \otimes a = \mathcal{C}_n(a \otimes b)$ is satisfied. Then, after computations:

$$X^{[2]}(t+1) = A_{22}(t)X^{[2]}(t) + S_2(t)X^{[2]}(t), \tag{3.24}$$

where:

$$A_{22}(t) = A^{[2]}(t) + B^{[2]}\eta_{2,t} + B_1^{[2]}(t)\xi_{c2,t} + B_2^{[2]}(t)\xi_{s2,t}, \tag{3.25}$$

and $S_2(t)$ is the following stochastic matrix:

$$\begin{aligned} S_2(t) = {} & B^{[2]}\big(N^2(t) - \eta_{2,t}\big) + B_1^{[2]}(t)\big(N_c^2(t) - \xi_{c2,t}\big) + B_2^{[2]}(t)\big(N_s^2(t) - \xi_{s2,t}\big) \\ & + M_1^2(n)\Big\{\big(A(t) \otimes B\big)N(t) + \big(A(t) \otimes B_1(t)\big)N_c(t) + \big(A(t) \otimes B_2(t)\big)N_s(t) \\ & + \big(B \otimes B_1(t)\big)N(t)N_c(t) + \big(B \otimes B_2(t)\big)N(t)N_s(t) + \big(B_1(t) \otimes B_2(t)\big)N_c(t)N_s(t)\Big\} \end{aligned} \tag{3.26}$$

In summary:

$$\mathcal{X}(t+1) = \mathcal{A}(t)\mathcal{X}(t) + \mathcal{N}_s(t), \tag{3.27}$$

with:

$$\mathcal{A}(t) = \begin{bmatrix} A(t) & O_{n \times n^2} \\ O_{n^2 \times n} & A_{22}(t) \end{bmatrix}; \tag{3.28}$$

$\mathcal{N}_s(t)$ is the following multiplicative noise:

$$\mathcal{N}_s(t) = \mathcal{S}(t)\mathcal{X}(t) = \begin{bmatrix}\mathcal{N}_{s1}(t) \\ \mathcal{N}_{s2}(t)\end{bmatrix} = \begin{bmatrix}S_1(t)\mathcal{X}_1(t) \\ S_2(t)\mathcal{X}_2(t)\end{bmatrix}, \tag{3.29}$$

with:

$$\mathcal{S}(t) = \begin{bmatrix} S_1(t) & O_{n\times n^2} \\ O_{n^2\times n} & S_2(t) \end{bmatrix}. \tag{3.30}$$

In order to compute the covariance matrices of $\mathcal{N}_s(t)$ and $\mathcal{N}_o(t)$, the definition of the *stack* operator is needed. The stack of a matrix A is the vector in $\mathbb{R}^{r\cdot c}$ that piles up all the columns of matrix A, and is denoted $\mathrm{s}t(A)$. The inverse operation is denoted $\mathrm{s}t^{-1}_{r,c}(\cdot)$, and transforms a vector of size $r\cdot c$ in an $r\times c$ matrix.

According to the multiplicative features of the noises, the computation of their covariance matrices requires the knowledge of the mean values of the Kronecker powers of the extended state up to degree 4. By denoting $Z_i(t) = \mathbb{E}\big[X^{[i]}(t)\big]$, $i = 1,\dots,4$, it comes after some Kronecker machineries:

$$\Psi_{s_{11}}(t) = \mathbb{E}\big[\mathcal{N}_{s1}(t)\mathcal{N}_{s1}^T(t)\big] = \mathrm{s}t^{-1}_{n,n}\Big(\mathbb{E}\big[S_1^{[2]}(t)\big]Z_2(t)\Big), \tag{3.31}$$

$$\Psi_{s_{12}}(t) = \mathbb{E}\big[\mathcal{N}_{s1}(t)\mathcal{N}_{s2}^T(t)\big] = \mathrm{s}t^{-1}_{n,n^2}\Big(\mathbb{E}\big[S_2(t)\otimes S_1(t)\big]Z_3(t)\Big), \tag{3.32}$$

$$\Psi_{s_{22}}(t) = \mathbb{E}\big[\mathcal{N}_{s2}(t)\mathcal{N}_{s2}^T(t)\big] = \mathrm{s}t^{-1}_{n^2,n^2}\Big(\mathbb{E}\big[S_2^{[2]}(t)\big]Z_4(t)\Big), \tag{3.33}$$

with:

$$\mathbb{E}\big[S_1^{[2]}(t)\big] = B^{[2]}\eta_{2,t} + B_1^{[2]}(t)\xi_{c2,t} + B_2^{[2]}(t)\xi_{s2,t}, \tag{3.34}$$

$$\begin{aligned}\mathbb{E}\big[S_2(t)\otimes S_1(t)\big] &= B_1^{[3]}(t)\xi_{c3,t} + \big(B_2^{[2]}(t)\otimes B_1(t)\big)\xi_{c1,s2,t}\\ &+\Big(\big(M_1^2(n)(A(t)\otimes B)\big)\otimes B\Big)\eta_{2,t} + \Big(\big(M_1^2(n)(A(t)\otimes B_1(t))\big)\otimes B_1(t)\Big)\xi_{c2,t}\\ &+\Big(\big(M_1^2(n)(A(t)\otimes B_2(t))\big)\otimes B_2(t)\Big)\xi_{s2,t} + \Big(\big(M_1^2(n)(B_1(t)\otimes B_2(t))\big)\otimes B_2(t)\Big)\xi_{c1,s2,t}\end{aligned} \tag{3.35}$$

$$\begin{aligned}\mathbb{E}\big[S_2^{[2]}(t)\big] &= B^{[4]}\big(\eta_{4,t}-\eta_{2,t}^2\big) + B_1^{[4]}(t)\big(\xi_{c4,t}-\xi_{c2,t}^2\big) + B_2^{[4]}(t)\big(\xi_{s4,t}-\xi_{s2,t}^2\big)\\ &+\big(M_1^2(n)\big)^{[2]}\Big\{\big(A(t)\otimes B\big)^{[2]}\eta_{2,t} + \big(A(t)\otimes B_1(t)\big)^{[2]}\xi_{c2,t} + \big(A(t)\otimes B_2(t)\big)^{[2]}\xi_{s2,t}\\ &+\big(B\otimes B_1(t)\big)^{[2]}\eta_{2,t}\xi_{c2,t} + \big(B\otimes B_2(t)\big)^{[2]}\eta_{2,t}\xi_{s2,t} + \big(B_1(t)\otimes B_2(t)\big)^{[2]}\xi_{c2,s2,t}\Big\}\\ &+M_1^2(n^2)\Big\{\big(B_1^{[2]}(t)\otimes B_2^{[2]}(t)\big)\big(\xi_{c2,s2,t}-\xi_{c2,t}\xi_{s2,t}\big)\\ &+\Big(B_1^{[2]}(t)\otimes\Big(M_1^2(n)\big(A(t)\otimes B_1(t)\big)\Big)\Big)\xi_{c3,t} + \Big(B_2^{[2]}(t)\otimes\Big(M_1^2(n)\big(A(t)\otimes B_1(t)\big)\Big)\Big)\xi_{c1,s2,t}\\ &+\Big(\Big(M_1^2(n)\big(A(t)\otimes B_2(t)\big)\Big)\otimes\Big(M_1^2(n)\big(B_1(t)\otimes B_2(t)\big)\Big)\Big)\xi_{c1,s2,t}\Big\}\end{aligned} \tag{3.36}$$

As far as the output noise covariance matrix, denote e_i, $i = 0,1,\dots,m$ the natural basis of $\mathbb{R}^{m+1}$. Then, matrix $H(t)$ may be written as:

$$H(t) = \sum_{i=1}^{m} e_i D_i(t) N_{o,i}(t), \tag{3.37}$$

and:

$$\begin{aligned}\Psi_o(t) = I\!E\big[\mathcal{N}_o(t)\mathcal{N}_o^T(t)\big] &= I\!E\big[H(t)\mathcal{X}_2(t)\mathcal{X}_2^T(t)H^T(t)\big] \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m} e_i D_i(t) I\!E\big[\mathcal{X}_2(t)\mathcal{X}_2^T(t)\big] D_j^T(t) e_j^T I\!E\big[N_{o,i}(t)N_{o,j}(t)\big].\end{aligned} \tag{3.38}$$

Recall that $\{N_{o,i}, N_{o,j}\}$ are uncorrelated for $i \neq j$, so that:

$$\Psi_o(t) = \sum_{i=1}^{m} e_i D_i(t) \cdot \mathrm{st}^{-1}_{n^2,n^2}\big(Z_4(t)\big) D_i^T(t) e_i^T I\!E\big[N'^2_{3,i}(t)\big]. \tag{3.39}$$

with:

$$I\!E\big[N'^2_{3,i}(t)\big] = \zeta_{i,4} - \zeta_{i,2}^2 + 4\rho_{i,t}^2\zeta_{i,2}. \tag{3.40}$$

4. The filtering algorithm

The position of the robot is given by the first two components of the augmented state, whose bilinear generation model, endowed with the output equation, is reported below for the ease of the reader:

$$\mathcal{X}(t+1) = \mathcal{A}(t)\mathcal{X}(t) + \mathcal{N}_s(t),$$

$$Y(t) = \mathcal{C}(t)\mathcal{X}(t) + \mathcal{V}(t) + \mathcal{N}_o(t).$$

The pair (x_t, y_t) is, then, estimated by means of the first two components of the augmented state estimate $\widehat{\mathcal{X}}(t)$. As is well known, the optimal choice for $\widehat{\mathcal{X}}(t)$ would be the conditional expectation w.r.t. all the Borel transformations of the measurements, whose computation in general cannot be obtained through algorithms of finite dimension. Nevertheless, from an applicative point of view, it is useful to look for finite-dimensional approximations of the optimal filter. In the present paper an implementable recursive filter is proposed, providing the optimal estimate w.r.t. the Hilbert space of all the affine (first-order polynomial) transformations of the output $Y(\tau)$, $\tau = 0, \ldots, t$ and performed as the projection of $\mathcal{X}(t)$ onto such a space [5]. As a consequence of the linear relationship between the pair (x_t, y_t) and $\mathcal{X}(t)$, it comes that the position estimate $(\hat{x}_t, \hat{y}_t)$ is the optimal linear estimate w.r.t. the output Y. The orientation angle estimate is provided, by means of the third and fourth components of $\widehat{\mathcal{X}}(t)$: $\hat{\theta}_t = \mathrm{atan2}\,(\hat{s}_t, \hat{c}_t)$.

According to the multiplicative feature of the noises, the computation of the covariance matrices of the noises $\mathcal{N}_s(t)$, $\mathcal{N}_o(t)$ requires the knowledge of the mean values of the Kronecker powers of the extended state up to degree 4 (as already reported in (3.31-3.33) and (3.39)). By defining $Z(t) = \big[Z_1^T(t)\ Z_2^T(t)\ Z_3^T(t)\ Z_4^T(t)\big]^T$, it is:

$$Z(t+1) = \mathcal{A}_Z(t)Z(t), \tag{4.1}$$

where:

$$\mathcal{A}_Z(t) = \begin{bmatrix} A(t) & O_{n\times n^2} & O_{n\times n^3} & O_{n\times n^4} \\ O_{n^2\times n} & A_{22}(t) & O_{n^2\times n^3} & O_{n^2\times n^4} \\ O_{n^3\times n} & O_{n^3\times n^2} & A_{33}(t) & O_{n^3\times n^4} \\ O_{n^4\times n} & O_{n^4\times n^2} & O_{n^4\times n^3} & A_{44}(t) \end{bmatrix}, \tag{4.2}$$

and

$$A_{33}(t) = A_{22}(t) \otimes A(t) + I\!E\big[S_2(t) \otimes S_1(t)\big], \qquad A_{44}(t) = A_{22}^{[2]}(t) + I\!E\big[S_2^{[2]}(t)\big]. \tag{4.3}$$

The initial condition $Z(0)$ has to be taken from the *a priori* knowledge concerning the initial state of the system. Assume x_0, y_0, θ_0 are independent Gaussian random variables, with mean value $\bar{x}_0$, $\bar{y}_0$, $\bar{\theta}_0$ and variance $\sigma_{x_0}^2$, $\sigma_{y_0}^2$, $\sigma_{\theta_0}^2$, respectively. From these statistics, by a direct computation and using the moments $\xi_{ci,sj,t}$, it is possible to derive $Z(0)$ and the covariance matrix $P_P(0) = \mathrm{Cov}\big(\mathcal{X}(0)\big)$ of the initial extended state vector.

The filter algorithm is below reported:

I) Compute the initial conditions, by means of $Z(0)$:

$$\widehat{\mathcal{X}}(0|-1) = \begin{bmatrix} Z_1(0) \\ Z_2(0) \end{bmatrix}, \qquad P_P(0) = \mathrm{Cov}\big(\mathcal{X}(0)\big), \qquad t = -1;$$

II) compute the output prediction:

$$\widehat{Y}(t+1|t) = \mathcal{C}(t+1)\widehat{\mathcal{X}}(t+1|t) + \mathcal{V}(t+1);$$

III) compute the output noise covariance matrix $\Psi_o(t+1)$ by means of (3.39) and (3.40);

IV) compute the Kalman gain and error covariance:

$$K(t+1) = P_P(t+1)\mathcal{C}^T(t+1)\Big(\mathcal{C}(t+1)P_P(t+1)\mathcal{C}^T(t+1) + \Psi_o(t+1)\Big)^{\dagger},$$

$$P(t+1) = \Big(I_{n+n^2} - K(t+1)\mathcal{C}(t+1)\Big)P_P(t+1);$$

where † denotes the Moore-Penrose pseudo-inverse matrix;

V) compute the estimate $\widehat{\mathcal{X}}(t+1)$:

$$\widehat{\mathcal{X}}(t+1) = \widehat{\mathcal{X}}(t+1|t) + K(t+1)\big(Y(t+1) - \widehat{Y}(t+1|t)\big),$$

$$\widehat{X}(t+1) = \big[I_n \quad O_{n\times n^2}\big]\widehat{\mathcal{X}}(t+1),$$

VI) increment the counter: $t \mapsto t+1$;

VII) compute the prediction $\widehat{\mathcal{X}}(t+1|t)$:

$$\widehat{\mathcal{X}}(t+1|t) = \mathcal{A}(t)\widehat{\mathcal{X}}(t);$$

VIII) compute the state noise covariance matrix $\Psi_s(t)$ by means of (3.31-3.36);

IX) compute the error covariance of the one-step prediction:

$$P_P(t+1) = \mathcal{A}(t)P(t)\mathcal{A}^T(t) + \Psi_s(t);$$

X) compute $Z(t)$ by means of (4.1);

XI) go to Step **II)**.

5. Simulation results

In this section we illustrate the effectiveness of the proposed approach through computer simulations. The performances of our algorithm are compared with those of a standard EKF. Note that although both algorithms have the same predictor-corrector structure, the EKF exploits the linear approximation of the original system equations (2.1)–(2.3) and (3.1), while our filter is based on exact equations of the system in the form (3.7) and (3.27). A circular trajectory in a rectangular room is considered in the simulations (see Fig. 1). The robot moves along the circle in the counterclockwise direction at a constant speed starting from the asterisk. The results reported in this section refer to simulations performed with $K_\rho = 0.01\text{m}$, $K_\theta = 0.02\text{rad}^2/\text{m}$ (noise parameters in (2.4)). The motion is performed in $T = 1000$ steps. We assume that the odometry readings are available at each motion step, while the laser readings are available every N_s steps, with $N_s \geq 1$. The simulations have been performed with several values of N_s to illustrate the ability of the proposed method w.r.t. the EKF in recovering the real trajectory even for large values of N_s.

We have considered $N_s = 1$ in the simulations of Figs. 2, 3 and 4 while $N_s = 50$ in Figs. 5, 6 and 7. Figs. 2 and 5 display the robot trajectory estimated through the EKF, the proposed method and the odometry together with the actual trajectory. With $N_s = 1$ both the proposed approach and the EKF provide a very good estimate (Fig. 2). However, with $N_s = 50$ the proposed filter performs nearly exact corrections unlike the EKF, whose linear approximations become too rough when the estimation errors are too large (Fig. 5).

Figs. 3 and 6 report the position error (that is the euclidean distance between the real robot position and its estimate) as a function of the traveled distance when applying the EKF, the proposed method and the odometry for $N_s = 1$ and $N_s = 50$, respectively. The same comparison is reported considering the orientation error in Figs. 4 and 7.

Finally, in Figs. 8 and 9 the effect of an increasing N_s on the EKF and the proposed method is illustrated. In Fig. 8 each point represents the square root of the mean square position error $\left(\left(\frac{1}{T}\sum_t[(x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2]\right)^{1/2}\right)$ and is obtained as the average of 20 different simulations (with different noise realizations). Fig. 9 reports the same for the orientation.

It is important to stress that for values of $N_s > 50$, in some simulations the EKF estimate has diverged, while our approach has maintained a bounded error. The performance of the proposed method has been tested with many different noise parameters, both in the odometry and in the laser readings, and similar results have been obtained.

6. Conclusions

In this paper we introduced and discussed a new analytical solution to the localization problem. In contrast with other previous approaches able to deal with the system non linearities (Markov Localization, Monte Carlo Localization), our algorithm is not a numerical solution. In particular, by introducing a new set of variables for the dynamics and for the readings and suitably exploiting some results of the polynomial filtering approach [4], [5], [11] a bilinear description of the system is derived and the best affine estimator of the robot configuration is obtained.

Note that the algorithm only concerns with the estimation process. A complete algorithm to the localization problem obviously includes several other tasks: feature extraction, data association, data synchronization etc.

In order to evaluate the performance of our algorithm we carried out a comparison with the standard EKF. Simulations are more suitable for this comparison since the ground truth is

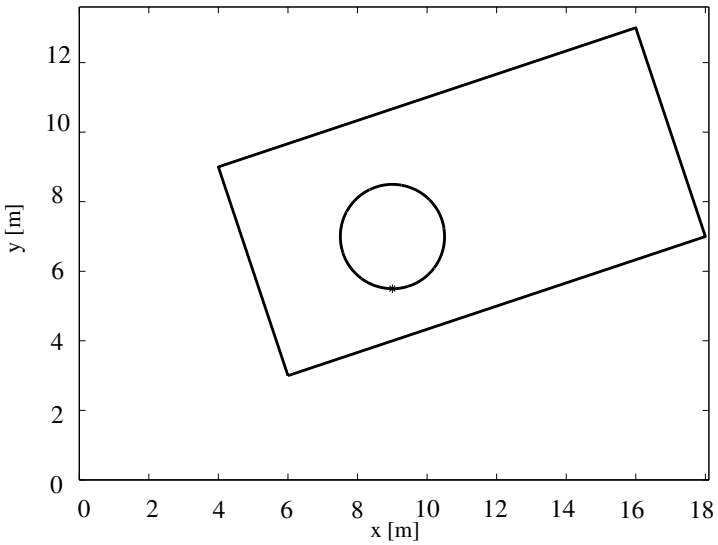

Figure 1: The environment and the actual robot trajectory

known and the comparison can be easily restricted to the estimation process, hence avoiding the effect due to the performance of the other tasks running simultaneously in a real experiment.

The simulation results clearly show that our algorithm outperforms the standard EKF especially in critical situations (low sensor precision, low data frequency).

This paper is a first step towards a novel methodology to approach the estimation problem in robotics. First of all we are interested in including the proposed algorithm in a complete framework to perform the robot localization in a real scenario, as in [2]. It will be very interesting to extend the algorithm to the case of other exteroceptive sensors, like GPS. We also want to extend the approach to the SLAM problem, where the nonlinearity can lead to the map inconsistency [6]. Finally, robustness with respect to the kidnapping problem can be achieved by using the multiple hypothesis idea as was done in [1] for the EKF.

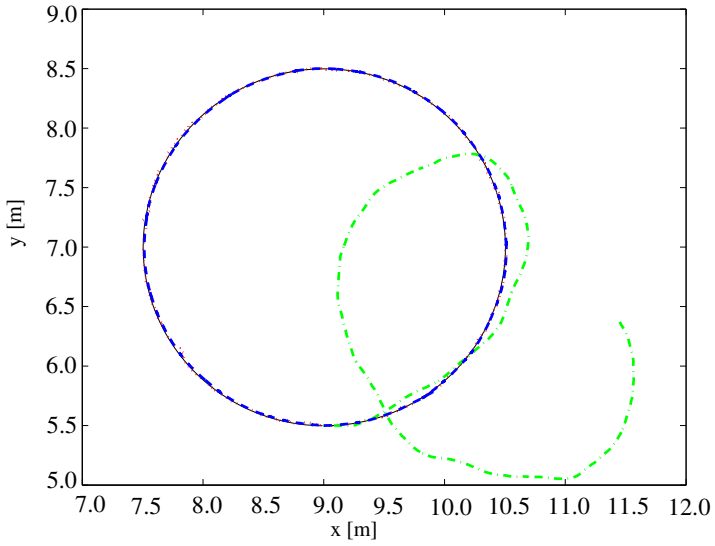

Figure 2: The real and the estimated trajectories with $N_s = 1$. Actual trajectory: black solid; odometry: green dot-dashed; EKF: blue dashed; the proposed filter: red dotted

Appendix

The moments $\xi_{ci,sj,t} = I\!E\left[N_c^i(t)N_s^j(t)\right]$, for $i, j = 1, \ldots, 4$, $2 \leq i + j \leq 4$ needed above in the algorithm can be computed with a direct integration obtaining:

$$
\begin{aligned}
\xi_{c2,t} &= \frac{1}{2} + \frac{1}{2}e^{-2\delta\rho_t^e K_\theta} - e^{-\delta\rho_t^e K_\theta}, \\
\xi_{s2,t} &= \frac{1}{2} - \frac{1}{2}e^{-2\delta\rho_t^e K_\theta}, \\
\xi_{c3,t} &= \frac{1}{4}e^{-\frac{9}{2}\delta\rho_t^e K_\theta} - \frac{3}{4}e^{-\frac{1}{2}\delta\rho_t^e K_\theta} + 2e^{-\frac{3}{2}\delta\rho_t^e K_\theta} \\
&- \frac{3}{2}e^{-\frac{5}{2}\delta\rho_t^e K_\theta}, \\
\xi_{c1,s2,t} &= -\frac{1}{4}e^{-\frac{1}{2}\delta\rho_t^e K_\theta} - \frac{1}{4}e^{-\frac{9}{2}\delta\rho_t^e K_\theta} + \frac{1}{2}e^{-\frac{5}{2}\delta\rho_t^e K_\theta}, \\
\xi_{c4,t} &= \frac{1}{8}e^{-8\delta\rho_t^e K_\theta} - \frac{5}{2}e^{-2\delta\rho_t^e K_\theta} + \frac{3}{8} - e^{-5\delta\rho_t^e K_\theta} \\
&+ 3e^{-3\delta\rho_t^e K_\theta}, \\
\xi_{c2,s2,t} &= \frac{1}{8} - \frac{1}{8}e^{-8\delta\rho_t^e K_\theta} + \frac{1}{2}e^{-5\delta\rho_t^e K_\theta} - \frac{1}{2}e^{-3\delta\rho_t^e K_\theta}, \\
\xi_{s4,t} &= \frac{1}{8}e^{-8\delta\rho_t^e K_\theta} - \frac{1}{2}e^{-2\delta\rho_t^e K_\theta} + \frac{3}{8},
\end{aligned}
$$

References

[1] K.O.Arras, J.A.Castellanos and R.Siegwart, "Feature-Based Multi-Hypothesis Localization and Tracking for Mobile Robots Using Geometric Constraints", In Proceedings of the 2002 IEEE International Conference on Robotics and Automation Washington, USA, May 2002.

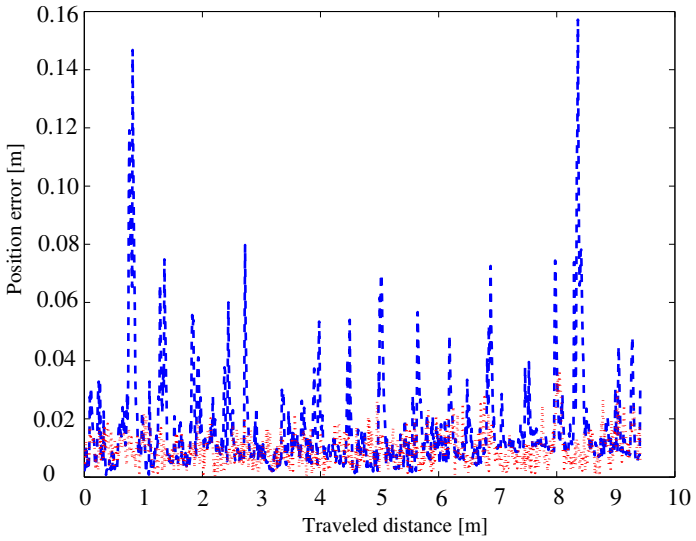

Figure 3: The position error with $N_s = 1$ vs the traveled distance. EKF: blue dashed; the proposed filter: red dotted (the odometry, omitted, is like in Fig. 6)

[2] K.O.Arras, N.Tomatis, B.T.Jensen and R.Siegwart, "Multisensor on-the-fly localization: Precision and reliability for applications", *Robotics and Autonomous Systems* 34, pp. 131–143, 2001.

[3] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. "Estimating the absolute position of a mobile robot using position probability grids". In Proceedings of the Thirteenth 44 National Conference on Artificial Intelligence, Menlo Park, August 1996. AAAI, AAAI Press/MIT Press.

[4] F. Carravetta, A. Germani, M. Raimondi. "Polynomial filtering for linear discrete time non-Gaussian systems". SIAM J. Control and Optimization, Vol. 34, No. 5, pp. 1666–1690, 1996;

[5] F. Carravetta, A. Germani, M. Raimondi. "Polynomial filtering of discrete time linear systems with multiplicative state noise". IEEE Transactions on Automatic Control, Vol. 42, No. 8, pp. 1106–1126, 1997;

[6] Castellanos, J. A., Neira, J. and Tardòs, J.D. (2004). "Limits to the consistency of the EKF-based SLAM". in Inteligent Autonomous Vehicles (IAV 2004), Lisbon, Portugal, July 2004.

[7] Chenavier F. and Crowley, J.L., (1992), "Position Estimation for a Mobile Robot using Vision and Odometry". IEEE International Conference on Robotics and Automation (ICRA), Nice, France, 1992

[8] Crowley, J.L., (1989). "World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging". IEEE International Conference on Robotics and Automation (ICRA), Scottsdale, AZ, USA, 1989.

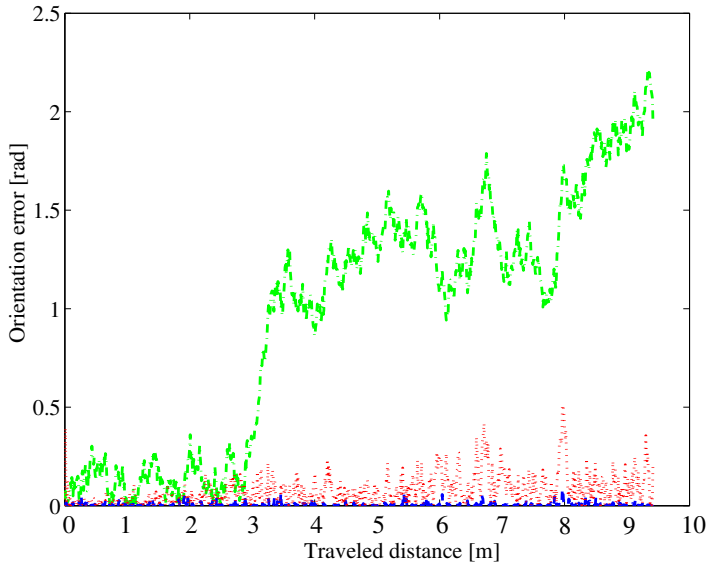

Figure 4: The orientation error with $N_s = 1$ vs the traveled distance. Odometry: green dot-dashed; EKF: blue dashed; the proposed filter: red dotted

[9] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors. "Sequential Monte Carlo Methods In Practice". Springer Verlag, New York, 2001.

[10] D. Fox, W. Burgard, and S. Thrun. "Markov localization for mobile robots in dynamic environments". Journal of Artificial Intelligence Research, 11:391-427, 1999.

[11] Alfredo Germani, Costanzo Manes and Pasquale Palumbo, "Polynomial Extended Kalman Filter", IEEE Transactions on Automatic Control, Vol. 50, No. 12, pp. 2059–2064, 2005;

[12] L.P. Kaelbling, A.R. Cassandra, and J.A. Kurien. "Acting under uncertainty: Discrete bayesian models for mobile-robot navigation". In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1996.

[13] J. Liu and R. Chen. "Sequential Monte Carlo methods for dynamic systems". Journal of the American Statistical Association, 93:1032-1044, 1998.

[14] A. Martinelli, "The odometry error of a mobile robot with synchronous drive system", IEEE Transactions on Robotics and Automation, Vol. 18, No. 3, pp. 399–405, June 2002.

[15] J. Neira, J.D. Tardos, J. Horn, G. Schmidt, "Fusing range and intensity images for mobile robot localization", IEEE Transactions on Robotics and Automation, Vol 15, No.1, 1999.

[16] M. Pitt and N. Shephard. "Filtering via simulation: auxiliary particle filter". Journal of the American Statistical Association, 94:590-599, 1999.

[17] R. Simmons and S. Koenig. "Probabilistic robot navigation in partially observable environments". In Proceedings of IJCAI-95, p. 1080-1087, Montreal, Canada, August 1995. IJCAI, Inc.

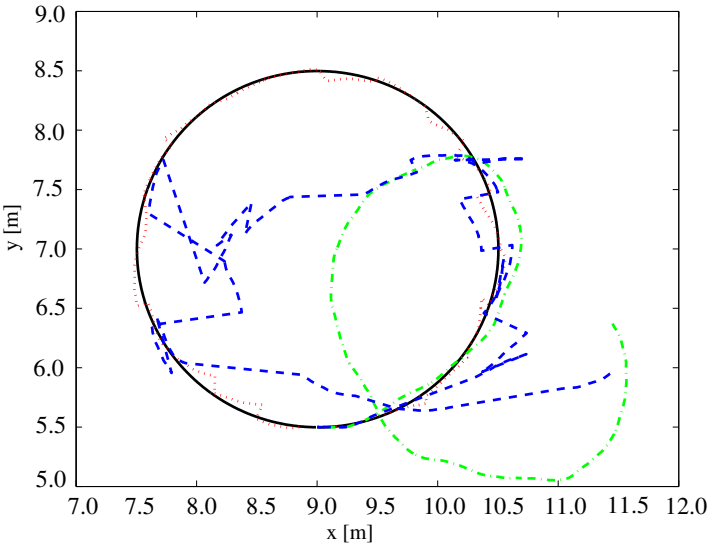

Figure 5: The real and the estimated trajectories with $N_s = 50$. Actual trajectory: black solid; odometry: green dot-dashed; EKF: blue dashed; the proposed filter: red dotted

[18] S. Thrun. "Bayesian landmark learning for mobile robot localization". Machine Learning, 33(1), 1998.

[19] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. "Robust Monte Carlo localization for mobile robots". Artificial Intelligence, 101:99–141, 2000

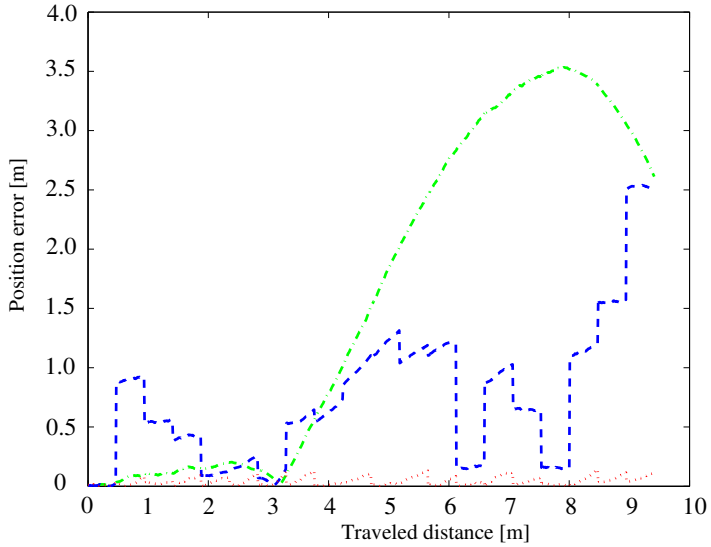

Figure 6: The position error with $N_s = 50$ vs the traveled distance. Odometry: green dot-dashed; EKF: blue dashed; the proposed filter: red dotted

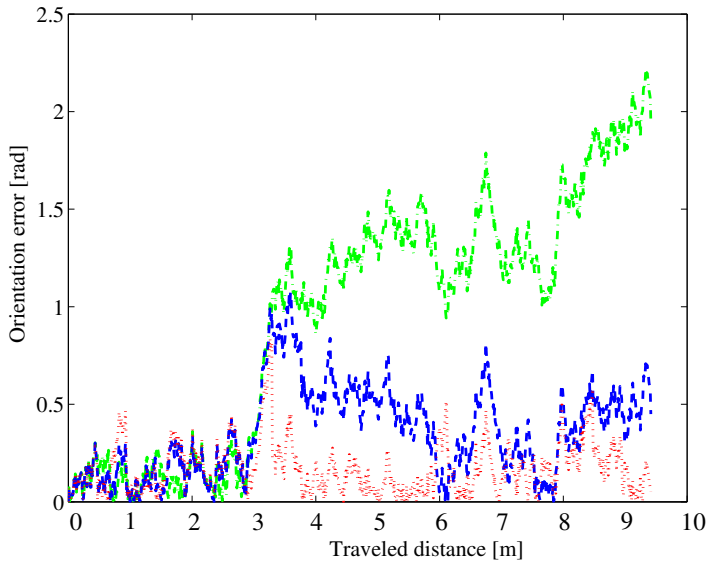

Figure 7: The orientation error with $N_s = 50$ vs the traveled distance. Odometry: green dot-dashed; EKF: blue dashed; the proposed filter: red dotted

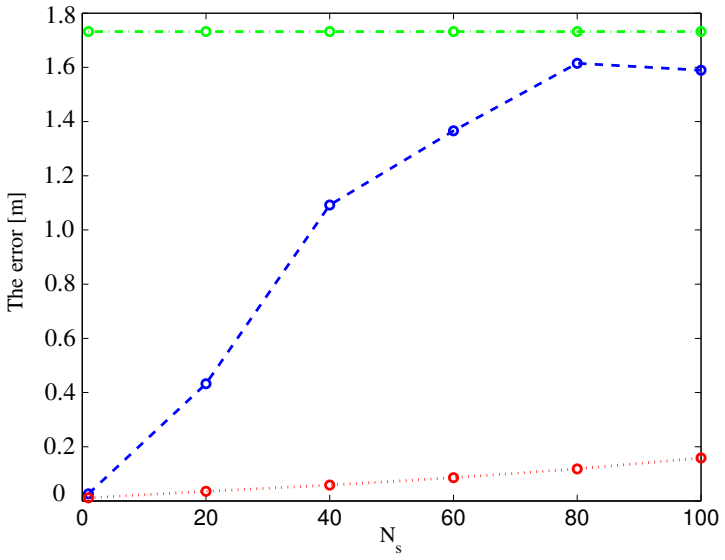

Figure 8: The square root of the mean square position error vs N_s. Odometry: green dot-dashed; EKF: blue dashed; the proposed filter: red dotted

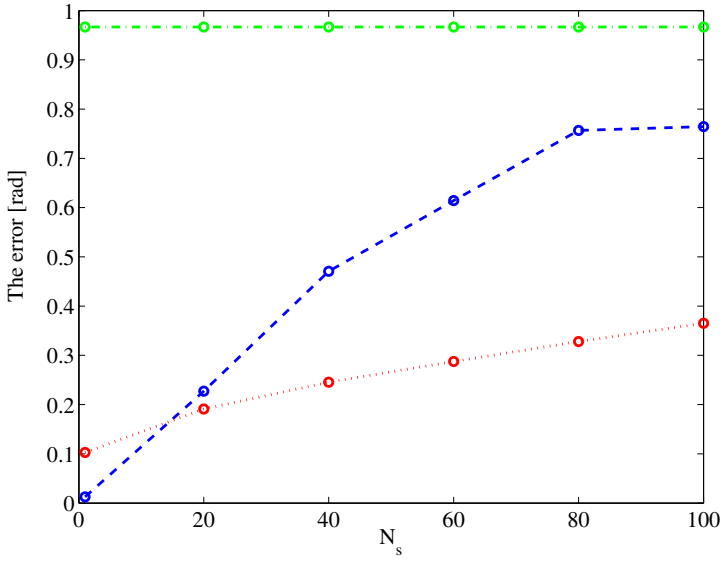

Figure 9: The square root of the mean square orientation error vs N_s. Odometry: green dot-dashed; EKF: blue dashed; the proposed filter: red dotted