

Polynomial Extended Kalman Filter in a SLAM framework

François Chanier, Paul Checchin, Christophe Blanc and Laurent Trassoudaine
 Laboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique
 Université de Clermont-Ferrand, 24, avenue des Landais,
 63177 Aubière cedex, France
 surname.name@lasmea.univ-bpclermont.fr

Abstract—This paper introduces an implementation of the Polynomial Extended Kalman Filter (PEKF) to solve the Simultaneous Localization and Map building (SLAM) problem. The proposed solution is a filtering algorithm which is a polynomial transformation of state evolution and measurement equations. The performances of the algorithm have been evaluated through two simulation runs. The first ones underline consistency improvement in comparison with the standard Extended Kalman Filter. The other simulation results show the PEKF efficiency when the values of measurement noises are high. At the end, experiments with Victoria Park data are presented too.

Index Terms—Simultaneous Localization and Mapping (SLAM), Polynomial Extended Kalman Filter (PEKF), consistency.

I. INTRODUCTION

The Simultaneous Localization and Mapping (SLAM) problem can be limited to asking the question: “*is it possible for a mobile robot to be placed at an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously determining its within this map?*” [1]. Since the seminal paper [2], the understanding of the Extended Kalman Filter (EKF) approach to the SLAM problem made real progress [1] [3].

Convergence properties and lower bound on the position uncertainty were provided with linear formulations of the SLAM problem [4]. However, apart from very few cases, both the dynamics of a mobile robot and the observation equations are nonlinear functions. When using EKF, these functions are linearized around the current estimated state. So, it introduces an approximation which can lead to divergence [5] [6] [7] [8]. In particular, Bailey *et al.* [7] prove with experiments and computer simulations that Unscented Kalman filter [9] and Iterated Extended Kalman filter [10] do not solve the nonlinear problem of SLAM.

Polynomial Extended Kalman Filter (PEKF) is a polynomial version of the EKF for the state estimation of non linear discrete-time stochastic systems. It was introduced for the first time by Carravetta *et al.* [11] in the framework of linear filtering. Since, it was applied to estimate parameters of bilinear systems [12] and parameters of non linear systems [13]. In each study case, the polynomial filter gives better simulation results than Unscented Kalman filter [9] and Iterated Extended Kalman filter [10]. This methodology was recently applied in vehicle localization with an a priori known map [14]. The simulation results showed in this article confirm the observations of Germani *et al.* [13]: the PEKF using

Carleman approximation [15] gives more accurate results than the standard EKF with nonlinear systems.

This paper presents a SLAM process using Polynomial extended Kalman Filter (PEKF SLAM) which was not applied before in this context, at our knowledge. The goal here is to ascertain if this application improves the SLAM results in the point of view of the accuracy and/or the estimate consistency. In order to do that, an experimental assessment is provided using the observations and the consistency criteria introduced in [8] and [7]. First, Huang *et al.* give both theoretical and practical proofs of the EKF SLAM inconsistencies. Using the same basic scenarios in the nonlinear two-dimensional problem, our paper shows that the PEKF SLAM approach stays consistent when uncertainties on vehicle states are initialized with high values. A second simulation run based on Monte Carlo tests [7] characterizes the filter behavior in a loop closing context. These experiments enable to check estimate consistency in comparison with EKF one. The simulation results confirm an improvement during the loop closing of the trajectory when the PEKF is applied.

The remainder of this paper is organized as follows: Section II describes the application of PEKF in a SLAM framework. In Section III, simulations bring a comparison between EKF SLAM and PEKF SLAM that shows improvements with regard to the observations presented in [7] and [8]. These simulation results give an assessment of PEKF estimate consistency. Finally, experimental results validate the proposed approach for large scale outdoor environment in Section IV.

II. PEKF SLAM IMPLEMENTATION

In this section, the SLAM nonlinear equations are first stated using an extended vehicle state vector. Second, the Carleman approximation of these states is introduced in order to apply the PEKF to *WorldCentric* SLAM framework.

A. SLAM WorldCentric formulation

1) *System state vectors*: In such approach, the global state \mathbf{X} consists in the concatenation of \mathbf{X}_R (vehicle states) and \mathbf{X}_m (map states):

$$\mathbf{X} = [\mathbf{X}_R \ \mathbf{X}_m]^T \quad (1)$$

The vector \mathbf{X}_m is composed by n features and is written as:

$$\mathbf{X}_m = [\ \mathbf{X}_{G_1} \ \cdots \ \mathbf{X}_{G_i} \ \cdots \ \mathbf{X}_{G_n} \]^T \quad (2)$$

The vector of the i^{th} feature is $\mathbf{X}_{G_i} = [x_i, y_i]^T$.

Vehicle states are represented by the vector \mathbf{X}_R :

$$\mathbf{X}_R = [x_v \ y_v \ \theta_v]^T \quad (3)$$

x_v and y_v are the Cartesian coordinates and θ_v is the vehicle orientation referenced to the frame attached to the initial vehicle pose.

2) *Extended vehicle state dynamics*: The robot \mathbf{X}_R moves in a 2D environment. This displacement is based on Ackerman model:

$$\begin{aligned} x_v(k+1) &= x_v(k) + \Delta t \cdot v \cdot \cos(\theta_v(k)) \\ y_v(k+1) &= y_v(k) + \Delta t \cdot v \cdot \sin(\theta_v(k)) \\ \theta_v(k+1) &= \theta_v(k) + \frac{\Delta t \cdot v \cdot \tan(\varphi)}{L} \end{aligned} \quad (4)$$

The forward velocity v and the steering wheels angle φ constitute the control vector $\mathbf{u} = [v, \varphi]^T$. The vehicle parameter L represents the distance between the two wheel axles.

In order to apply the PEKF SLAM, the vehicle state vector has to be extended. Indeed, in the spatial relationships [2] linking vehicle pose to the feature poses, it is not the angle θ_v which is directly used but it is the cosine and the sine of this angle. To transform these nonlinear equations in polynomial equations, the vehicle state vector is extended:

$$\mathbf{Z}_R = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} x_v \\ y_v \\ \sin(\theta_v) \\ \cos(\theta_v) \end{bmatrix} \quad (5)$$

To be clear in the sequel, a new control vector based on vector \mathbf{u} is defined:

$$\mathbf{U} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} v \cdot \Delta t \\ \sin\left(\frac{\Delta t \cdot v \cdot \tan(\varphi)}{L}\right) \\ \cos\left(\frac{\Delta t \cdot v \cdot \tan(\varphi)}{L}\right) \end{bmatrix} \quad (6)$$

With new vectors (5) and (6), the extended state dynamics is given by the equations:

$$\hat{\mathbf{Z}}_R(k+1) = \begin{bmatrix} z_1(k) + z_4(k) \cdot u_1 \\ z_2(k) + z_3(k) \cdot u_1 \\ z_3(k) \cdot u_3 + z_4(k) \cdot u_2 \\ z_4(k) \cdot u_3 - z_3(k) \cdot u_2 \end{bmatrix} \quad (7)$$

These equations are rewritten under a matrix formulation:

$$\hat{\mathbf{Z}}_R(k+1) = \mathbf{A}(k) \cdot \hat{\mathbf{Z}}_R(k) \quad (8)$$

$$\text{with } \mathbf{A}(k) = \begin{bmatrix} 1 & 0 & 0 & u_1 \\ 0 & 1 & u_1 & 0 \\ 0 & 0 & u_3 & u_2 \\ 0 & 0 & -u_2 & u_3 \end{bmatrix} \quad (9)$$

The covariance matrix associated to the state vector (5) is obtained with:

$$\mathbf{P}_R(k+1) = \mathbf{A}(k) \cdot \mathbf{P}_R(k) \cdot \mathbf{A}(k)^T + \mathbf{J}(k) \cdot \mathbf{Q}_U \cdot \mathbf{J}(k)^T \quad (10)$$

\mathbf{Q}_U is the noise matrix associated to the control vector \mathbf{U} (6) and $\mathbf{J}(k)$ is the Jacobian of \mathbf{Z}_R (7) with respect to \mathbf{U} :

$$\mathbf{J}(k) = \begin{bmatrix} z_4(k) & 0 & 0 \\ z_3(k) & 0 & 0 \\ 0 & z_4(k) & z_3(k) \\ 0 & -z_3(k) & z_4(k) \end{bmatrix} \quad (11)$$

The system vector (1) is now written as:

$$\mathbf{Z} = (\mathbf{Z}_R \ \mathbf{X}_m)^T \quad (12)$$

B. Carleman approximation of WorldCentric SLAM system states

Carleman approximation of a second order [15] is realized to the presented SLAM system. It is achieved by defining the sequence $\mathbf{Z}^{[2]}$ of the Kronecker powers of the states of the vector \mathbf{Z} (12). As the vector \mathbf{Z} is composed by the vehicle state vector and the map state vector, the final vector \mathcal{Z} can be written as:

$$\mathcal{Z} = [\mathbf{Z} \ \mathbf{Z}^{[2]}]^T = [\mathbf{Z}_R \ \mathbf{Z}_R^{[2]} \ \mathbf{X}_m \ \mathbf{X}_m^{[2]}]^T \quad (13)$$

As all the states of the new vector \mathcal{Z} are not used in observation and landmark initialization equations, reductions have been done to lighten the formulations. In the following, the Kronecker powers of vehicle state vector (5) and feature state vector (2) will be introduced.

1) *Kronecker powers apply to the vehicle states*: The Kronecker powers of \mathbf{Z}_R with the states described by equations (5) is:

$$\mathbf{Z}_R = \begin{bmatrix} \mathbf{Z}_R \\ \mathbf{Z}_R^{[2]} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \\ z_9 \\ z_{10} \\ z_{11} \end{bmatrix} = \begin{bmatrix} x_v \\ y_v \\ \sin(\theta_v) \\ \cos(\theta_v) \\ \sin^2(\theta_v) \\ \cos^2(\theta_v) \\ x_v \cdot \sin(\theta_v) \\ x_v \cdot \cos(\theta_v) \\ y_v \cdot \sin(\theta_v) \\ y_v \cdot \cos(\theta_v) \\ \sin(\theta_v) \cdot \cos(\theta_v) \end{bmatrix} \quad (14)$$

2) *Kronecker powers apply to the feature states*: The Kronecker power application to the feature state vector (2) requires to take into account vehicle states (14). Variables multiplying vehicle states to feature states are created. The new vector of the i^{th} feature is:

$$\mathcal{Z}_i = \begin{bmatrix} \mathbf{X}_{G_i} \\ \mathbf{X}_{G_i}^{[2]} \end{bmatrix} = \begin{bmatrix} z_{i1} \\ z_{i2} \\ z_{i3} \\ z_{i4} \\ z_{i5} \\ z_{i6} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ \sin(\theta_v) \cdot x_i \\ \cos(\theta_v) \cdot x_i \\ \sin(\theta_v) \cdot y_i \\ \cos(\theta_v) \cdot y_i \end{bmatrix} \quad (15)$$

C. PEKF SLAM formulation

As the PEKF is a Kalman filter, the PEKF SLAM approach uses the same algorithm presented in [4]. In this paper, only the equations of the vehicle dynamics and the observation are described. These modifications are the consequences of the restating of the system state vectors (Section II-B).

1) *Equations of vehicle state evolution:* The new vehicle state vector \mathcal{Z}_R (14) evolves with the equations (7):

$$\hat{\mathcal{Z}}_R(k+1) = \begin{bmatrix} z_1(k) + u_1 \cdot z_4(k) \\ z_2(k) + u_1 \cdot z_3(k) \\ z_3(k) \cdot u_3 + z_4(k) \cdot u_2 \\ z_4(k) \cdot u_3 - z_3(k) \cdot u_2 \\ z_5(k) \cdot (u_3)^2 + z_{11}(k) \cdot 2 \cdot u_2 \cdot u_3 \\ \quad + z_6(k) \cdot (u_2)^2 \\ z_6(k) \cdot (u_3)^2 - z_{11}(k) \cdot 2 \cdot u_2 \cdot u_3 \\ \quad + z_5(k) \cdot (u_2)^2 \\ z_7(k) \cdot u_3 + z_8(k) \cdot u_2 \\ \quad + z_6(k) \cdot u_1 \cdot u_2 + z_{11}(k) \cdot u_1 \cdot u_3 \\ z_8(k) \cdot u_3 - u_2 \cdot z_7(k) \\ \quad + z_6(k) \cdot u_1 \cdot u_3 - z_{11}(k) \cdot u_1 \cdot u_2 \\ z_9(k) \cdot u_3 + u_2 \cdot z_{10}(k) \\ \quad + z_5(k) \cdot u_1 \cdot u_3 + z_{11}(k) \cdot u_1 \cdot u_2 \\ z_{10}(k) \cdot u_3 - u_2 \cdot z_9(k) \\ \quad + z_{11}(k) \cdot u_1 \cdot u_3 - z_5(k) \cdot u_1 \cdot u_2 \\ z_{11}(k) \cdot (u_3)^2 - u_2 \cdot u_3 \cdot z_5(k) \\ \quad + z_6(k) \cdot u_2 \cdot u_3 - z_{11}(k) \cdot (u_2)^2 \end{bmatrix} \quad (16)$$

The calculation of the covariance matrix associated to vehicle state vector (16) is:

$$\mathbf{P}_{\mathcal{Z}_R}(k+1) = \mathbf{J}_{\mathcal{Z}_R} \cdot \mathbf{P}_{\mathcal{Z}_R}(k) \cdot \mathbf{J}_{\mathcal{Z}_R}^T + \mathbf{J}_U \cdot \mathbf{Q}_U \cdot \mathbf{J}_U^T \quad (17)$$

$\mathbf{J}_{\mathcal{Z}_R}$ is the Jacobian of the equations (16) with respect to $\hat{\mathcal{Z}}_R$, \mathbf{J}_U is the Jacobian of the equations (16) with respect to \mathbf{U} .

2) *Equations of feature state evolution:* Even if the environment is supposed to be static, with the new formulation of feature vector (15), some states of \mathcal{Z}_i have to be modified during the prediction step. This evolution is given by:

$$\hat{\mathcal{Z}}_i(k+1) = \begin{bmatrix} z_{i1}(k) \\ z_{i2}(k) \\ z_{i3}(k) \cdot u_3 + z_{i4}(k) \cdot u_2 \\ z_{i4}(k) \cdot u_3 - z_{i3}(k) \cdot u_2 \\ z_{i5}(k) \cdot u_3 + z_{i6}(k) \cdot u_2 \\ z_{i6}(k) \cdot u_3 - z_{i5}(k) \cdot u_2 \end{bmatrix} \quad (18)$$

The calculation of the covariance matrix associated to feature state vector (18) is:

$$\mathbf{P}_{\mathcal{Z}_i}(k+1) = \mathbf{J}_{\mathcal{Z}_i} \cdot \mathbf{P}_{\mathcal{Z}_i}(k) \cdot \mathbf{J}_{\mathcal{Z}_i}^T + \mathbf{J}_{U_i} \cdot \mathbf{Q}_U \cdot \mathbf{J}_{U_i}^T \quad (19)$$

$\mathbf{J}_{\mathcal{Z}_i}$ is the Jacobian of the equations (18) with respect to $\hat{\mathcal{Z}}_i$, \mathbf{J}_{U_i} is the Jacobian of the equations (18) with respect to \mathbf{U} .

3) *Observation equations:* In *WorldCentric* SLAM approach, observation function \mathbf{h}_i that formulates the global coordinates of the feature \mathbf{X}_{G_i} in robot frame R is given by:

$$\mathbf{X}_{G_i}^R = \mathbf{h}_i(\mathbf{X}_R, \mathbf{X}_{G_i}) + \mathbf{w}_i(k+1) \quad (20)$$

$\mathbf{w}_i(k+1)$ is the noise on the feature measurements. The development of this equation in function of the vehicle states and the feature states, equations (4) and (2), is:

$$\hat{\mathbf{X}}_{G_i}^R(k+1) = \ominus \hat{\mathbf{X}}_R \oplus \hat{\mathbf{X}}_{G_i} \quad (21)$$

$$= \begin{bmatrix} x_i \cdot \cos(\theta_v) + y_i \cdot \sin(\theta_v) - x_v \cdot \cos(\theta_v) - y_v \cdot \sin(\theta_v) \\ y_i \cdot \cos(\theta_v) - x_i \cdot \sin(\theta_v) + x_v \cdot \sin(\theta_v) - y_v \cdot \cos(\theta_v) \end{bmatrix} \quad (22)$$

With the extended state vector of the i^{th} (18) and the vehicle (14), the observation equations is rewritten:

$$\hat{\mathcal{Z}}_{G_i}^R = \begin{bmatrix} z_{i4} + z_{i5} - z_8 - z_9 \\ -z_{i3} + z_{i6} + z_7 - z_{10} \end{bmatrix} \quad (23)$$

This output equation can now be seen now as a linear transformation of the augmented state \mathcal{Z} defined by (1).

4) *Equations of the feature mapping:* The landmark initialization step corresponds to the function \mathbf{g} . It returns the vector $\mathbf{X}_{G_{new}}$ referenced in the frame attached to the global map:

$$\mathbf{X}_{G_{new}} = \mathbf{g}(\mathbf{X}_R, \mathbf{X}_{G_{new}}^R) + \mathbf{w}_{new}(k+1) \quad (24)$$

Measurement vector of a new feature $\mathbf{X}_{G_{new}}^R$ is composed with the coordinates x_{new} and y_{new} referenced in the frame attached to the vehicle R . $\mathbf{w}_{new}(k+1)$ is the measurement noise of new feature. The landmark initialization function \mathbf{g} is an compounding operation as defined in [2]. The vector $\mathbf{X}_{G_{new}}$ can be written as a function of the vehicle states (14) and the new feature states:

$$\hat{\mathbf{X}}_{G_{new}} = \hat{\mathcal{Z}}_R \oplus \hat{\mathbf{X}}_{G_{new}} \quad (25)$$

$$\hat{\mathcal{Z}}_{G_{new}} = \begin{bmatrix} x_{new} \cdot z_4 - y_{new} \cdot z_3 + z_1 \\ x_{new} \cdot z_3 + y_{new} \cdot z_4 + z_2 \end{bmatrix} \quad (26)$$

To stay consistent with the feature vector (15), the Kronecker powers are applied to the equations (26):

$$\hat{\mathcal{Z}}_{G_{new}} = \begin{bmatrix} x_{new} \cdot z_4 - y_{new} \cdot z_3 + z_1 \\ x_{new} \cdot z_3 + y_{new} \cdot z_4 + z_2 \\ x_{new} \cdot z_{11} - y_{new} \cdot z_5 + z_7 \\ x_{new} \cdot z_6 - y_{new} \cdot z_{11} + z_8 \\ x_{new} \cdot z_5 + y_{new} \cdot z_{11} + z_9 \\ x_{new} \cdot z_{11} + y_{new} \cdot z_6 + z_{10} \end{bmatrix} \quad (27)$$

The system covariance matrix after the adding of the new feature is given by:

$$\mathbf{P}_{\mathcal{Z}} = \mathbf{G} \cdot \begin{bmatrix} \mathbf{P}_{\mathcal{Z}} & 0 \\ 0 & \mathbf{P}_{\mathcal{Z}_{G_{new}}} \end{bmatrix} \cdot \mathbf{G}^T \quad (28)$$

Jacobian matrix \mathbf{G} is:

$$\mathbf{G} = \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{G}_{\mathcal{Z}} & \mathbf{G}_{new} \end{bmatrix} \quad (29)$$

$\mathbf{G}_{\mathcal{Z}}$ is the Jacobian of the equations (27) with respect to $\hat{\mathcal{Z}}$, \mathbf{G}_{new} is the Jacobian of the equations (27) with respect to \mathbf{X}_{new} .

III. SIMULATION RESULTS

To qualify the PEKF results, the same simulations presented in [7] and [8] were performed. The first ones show the filter behaviors when vehicle uncertainty initialization changes in a very basic scenario. In the other ones, the consistency evolution is studied during a loop closing trajectory [7].

A. First simulations: uncertainty initialization

Two scenarios are studied: robot stationary and robot moves. In both cases, only one landmark is observed. The uncertainty theoretical limits are obtained by linearizing system equations around the ground truth solution (simulated data without noise).

In [8], it is proved that the inconsistency problem of EKF SLAM can be neglected if the initial uncertainty of the robot orientation is much smaller. In order to observe the estimate inconsistency, the initial values of vehicle state uncertainty are $\mathbf{P}_{\mathbf{x}_R}(0) = \text{diag}(1, 1, 0.03)$ for the EKF SLAM and $\mathbf{P}_{\mathbf{z}_R}(0) = \text{diag}(1, 1, 0.03, 1)$. The other simulation parameters are presented in [8].

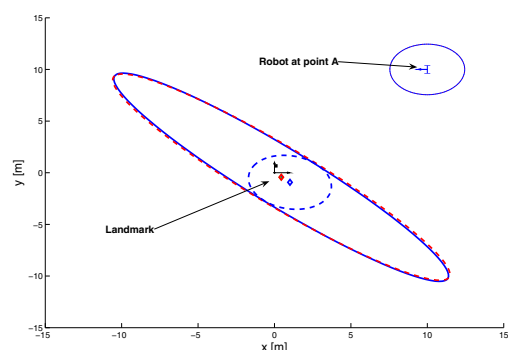


Fig. 1. The limits of landmark uncertainty when the robot is stationary and observes one landmark: EKF with ideal Jacobians (blue lines), EKF (blue dashed lines) and PEKF (red dashed lines)

1) *Robot stationary*: For the first simulation, the robot is stationary at point *A* and makes n observations of only one new landmark (Fig. 1). This figure is generated by performing 1000 updates assuming that only the range and bearing measurements are corrupted by random Gaussian noise (values are presented in Tab. I). The dead reckoning data are not corrupted.

In Fig. 1, it can be seen that the uncertainty EKF estimates of the landmark (blue dashed line) is reduced far below the theoretical limit (blue line). It demonstrates the inconsistency of EKF SLAM (*Theorem 4.1* in [8]). However, the uncertainty PEKF estimate (red dashed line) is equal to the theoretical one. It proves the estimate consistency of the PEKF SLAM.

In [8], it is proved that observing a single new landmark will not improve the knowledge of the robot orientation. It is right when the PEKF SLAM is used. Indeed, with the formulation of the PEKF SLAM, the observation equation (23) do not used the state with robot orientation, z_3 and z_4 . During the Kalman estimation step, only robot states, z_{10} , z_{11} , z_{12} and z_{13} are

modified. So, the landmark observation does not improve the knowledge of the robot orientation.

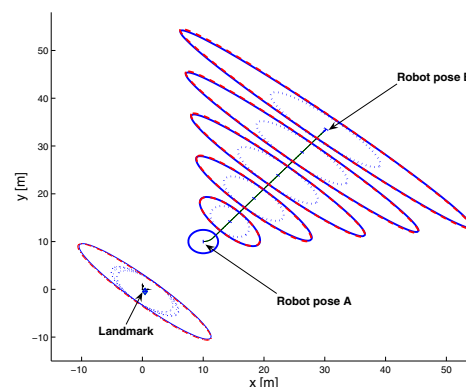


Fig. 2. The limits of landmark uncertainty when the robot moves and observes one landmark: EKF with ideal Jacobians (blue lines), EKF (blue dashed lines) and PEKF (red dashed lines)

2) *Robot moves*: For the second simulation, Fig. 2, the robot first keeps stationary at point *A* and makes observations 1000 times. The true Jacobians are used at point *A* to guarantee the consistency of the estimates before the robot moves. Then the robot moves 500 steps to *B* and keeps observing the same landmark while moving. The value of sensor noises are the same than in the first simulation. The control data, v and φ , are now corrupted by random Gaussian (noise values in Tab. I).

In Fig. 2, it can be seen that the uncertainty EKF estimates of both the landmark and the robot (blue dashed line) is reduced far below the theoretical limit (blue line) as soon as the robot starts to moving. It demonstrates the inconsistency of EKF SLAM (*Theorem 4.2* in [8]). As in Section III-A1, the uncertainty PEKF estimates (red dashed line) are consistent. It is clearly proves that only PEKF SLAM provides consistent uncertainty estimates.

3) *Simulation conclusion*: The common idea used in [5] [6] [7] [8] is that the consistency of SLAM estimate is evaluated based on the fact: “**Keep observing new landmarks does not help in reducing the robot pose uncertainty**”. The simulation results prove that the PEKF SLAM answer to this affirmation.

B. Second simulations: loop closing context

In the following experiments, the consistency analysis is made in a more complicated scenario. The behavior of the PEKF SLAM is compared to the EKF SLAM one in a loop closing context. The vehicle realizes two loops of the trajectory presented in [7]. As in the article of Bailey *et al.*, an average Normalized Estimation Error Squared (NEES) [10] over 50 Monte Carlo runs is calculated to measure the consistency of the estimates. The simulation parameters are summarized in Tab. I. For the PEKF SLAM, the orientation angle $\hat{\theta}_v$ is provided by means of the third and fourth components of the vector (14): $\hat{\theta}_v = \text{atan2}(Z(3), Z(4))$.

In Fig. 3, the NEES distance evolutions are presented when the EKF SLAM is used with low (around 10^{-3}) and high

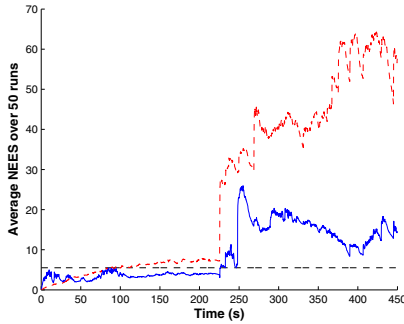


Fig. 3. Average NEES of the vehicle pose states over 50 Monte Carlo runs and 95% probability threshold for a 3-dimensional vector (dashed black line): EKF SLAM with high (dashed red curve) and low (blue curve) initial uncertainty values on vehicle states

($\mathbf{P}_{\mathbf{x}_R}(0) = \text{diag}(1, 1, 0.03)$) uncertainty initializations. In the first case (Fig. 3, blue curve), NEES values exceed the χ^2 threshold (dashed black line) when vehicle trajectory is closed at 220 s. It means that the EKF results are inconsistent. When the uncertainty on the vehicle state is initialized with high values, the inconsistency problem is more important during the first loop as shown in Fig. 3. After the loop closing, NEES values go past the threshold and reach lower values than the blue curve in Fig. 3. These results underline an important fact: during the loop closing, the EKF SLAM fails in every studied cases [7]. Moreover, an increase of initial values of uncertainty amplifies the inconsistency problem during the first loop as it is shown in [5] and [6].

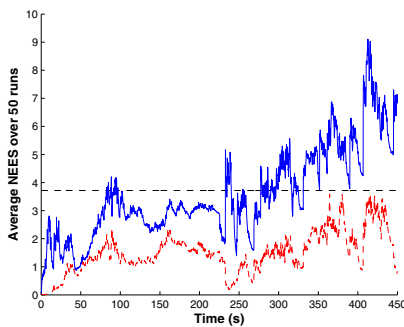


Fig. 4. Average NEES of the vehicle pose states over 50 Monte Carlo runs and 95% probability threshold for a 3-dimensional vector (dashed black line): PEKF SLAM with high (dashed red curve) and low (blue curve) initial uncertainty values on vehicle states

In Fig. 4, the NEES distance evolution using the PEKF SLAM application is presented. When the initial values of uncertainty are low (Fig. 4, blue curve), the PEKF SLAM estimates are consistent during the loop closing at the difference of the EKF SLAM. Indeed, the NEES average curve stays under the probability threshold and the strong increase observed in Fig. 3 at 220 s is not seen. Nevertheless, after the loop closing, the curve fluctuates with values higher than the threshold. The approximation due to the linearization in the prediction step (Section II-C1) begins to be too important.

If initial uncertainty is high (Fig. 4, dashed red curve), the estimates are also consistent during the loop closing like those with low initial uncertainty (Fig. 4, blue curve). Furthermore, the results is better as NEES values are lower. However, the same variations are observed at the end of the second loop. So, the uncertainty initialization has no effect on the PEKF behavior but reduces inconsistency problems at the difference of the EKF one. This fact corroborates the observations made in Section III-A.

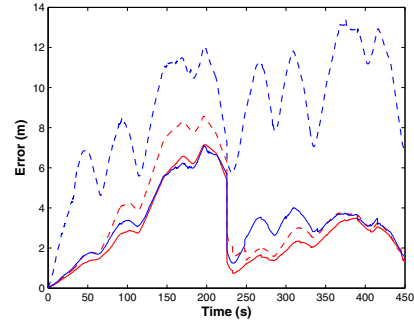


Fig. 5. Average error on the vehicle estimated trajectory over 50 Monte Carlo runs: EKF SLAM with high (dashed blue curve) and low (blue curve) initial uncertainty values on vehicle states and PEKF SLAM with high (dashed red curve) and low (red curve) initial uncertainty values on vehicle states

In Fig. 5, the evolutions of the average errors related to the vehicle poses over 50 Monte Carlo runs are presented for the four studied cases. PEKF (red curves) gives more accurate results than the EKF (blue curves) during the second loop of the trajectory. This fact confirms the consistency improvement. When using the PEKF, the increase of initial uncertainty makes the filter less accurate even if the estimates are more consistent (Fig. 4). Our next study will be to understand this phenomena in order to improve PEKF results.

The PEKF SLAM improves estimate consistency during the loop closing in each studied cases. However, even if the initialization of vehicle state uncertainty is good, it does not solve consistency problems of Kalman filters when vehicle realizes more than two or three loops.

IV. OUTDOOR PEKF SLAM

An experiment with a well-known benchmark dataset [16] is presented to validate the reliability of our PEKF SLAM implementation. The vehicle (a truck) is equipped with a 2D laser range-finder, a GPS sensor and dead reckoning sensors (forward velocity and steering angle). During the experiment, the vehicle moves in a 3 km path along the Victoria Park, in Sydney, Australia. The laser is used to detect and measure the position of trees (and not radius value). The poor quality of the dead reckoning data (the bumpy terrain introduced many unexpected perturbations) and the presence of many spurious (e.g. people) make the SLAM problem very interesting with this dataset. The standard deviations on the data are summarized in Tab. I.

In Fig. 6 the PEKF SLAM results are shown with the complete vehicle trajectory and the detected environmental

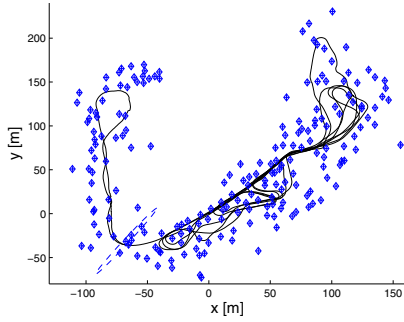


Fig. 6. PEKF SLAM experiment (Victoria Park dataset): Final 2D-point feature and vehicle estimated trajectory

features. The PEKF SLAM approach keeps a good level of accuracy during all the path. As vehicle trajectory estimate is accurate, the data association uses only a “simple” nearest neighbor algorithm.

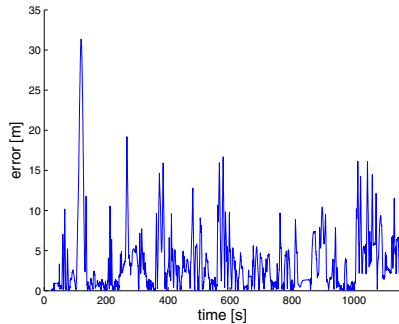


Fig. 7. PEKF SLAM experiment (Victoria Park dataset): Evolution of the error on vehicle estimated trajectory

In Fig. 7, the error on the estimates of the vehicle pose calculated with ground truth data is shown. High error values during time intervals can be observed and they corresponds to the losses of GPS signal. Except during these moments, the error is maintained under 2 meters. As the ground truth is a low cost GPS sensor with a standard deviation of 1 meter, these results are quite good. Furthermore, they prove that *WorldCentric* PEKF SLAM application on real data is reliable.

V. CONCLUSION

In this paper, a SLAM process using polynomial extended Kalman filter is presented. To introduce the PEKF SLAM formulation, the Carleman approximation of the nonlinear equations is described as soon as new state vectors.

In order to assess the PEKF SLAM performances, a comparison with the standard EKF is carried out based on two different simulation runs, respectively defined in [7] and [8]. The results of the first one underline that this application gives consistent uncertainty estimates (nearly equal to the theoretical ones) when initial uncertainty is important. The other simulations, based on 50 Monte Carlo runs, show that

	Noise values			
	Laser range-finder		Control	
	σ_ρ [m]	σ_β [rad]	σ_v [ms ⁻¹]	σ_φ [rad]
First simulations	0.1	$\pi/180$	0.02	$3.\pi/180$
Second simulations	0.1	$\pi/180$	0.3	$3.\pi/180$
Victoria Park Data	0.5	$\pi/60$	0.7	$7.\pi/180$

TABLE I
RANGE-BEARING AND CONTROL STANDARD DEVIATIONS

PEKF improves consistency of the results during the loop closing. The experimental results on the Victoria Park dataset attest that PEKF SLAM can be applied to a real context.

REFERENCES

- [1] T. Bailey and H. Durrant-Whyte, “Simultaneous Localisation and Mapping (SLAM) : Part II State of the Art,” *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 108–117, September 2006.
- [2] R. Smith, M. Self, and P. Cheeseman, “Estimating uncertain spatial relationships in robotics,” in *Proceedings, 2nd workshop on Uncertainty in Artificial Intelligence (AAAI)*, August 1986, pp. 1–21.
- [3] H. Durrant-Whyte and J. Guivant, “Simultaneous Localisation and Mapping (SLAM) : Part I The Essential Algorithms,” *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 99–110, June 2006.
- [4] M. Dissanayake, P. N. ans S. Clark, H. Durrant-Whyte, and M. Csorba, “A Solution to the simultaneous Localization and Map building (SLAM) problem,” *IEEE Transactions on Robotics and automation*, vol. 17, no. 3, pp. 229–241, June 2001.
- [5] S. Julier and K. Uhlmann, “A counter example to the theory of simultaneous localization and map building,” in *Proceedings, IEEE International Conference on Robotics and Automation*, Seoul, Korea, October 2001, pp. 4238–4243.
- [6] J. Castellanos, J. Neira, and J. Tardos, “Limits to the consistency of EKF-based SLAM,” in *Proceedings, IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon, Spain*, July 2004.
- [7] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, “Consistency of the EKF-SLAM Algorithm,” in *Proceedings, IEEE International Conference on Robots and Systems, Beijing, China*, 2006.
- [8] S. Huang and G. Dissanayake, “Convergence and Consistency Analysis for Extended Kalman Filter based SLAM,” *IEEE Transactions on Robotics and Automation*, vol. 21, pp. 1036–1049, October 2007.
- [9] S. Julier and J. Uhlmann, “A New Extension of the Kalman Filter to Nonlinear Systems,” in *Proceedings of AeroSense, 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, FL, USA*, October 1997.
- [10] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley and SONS, INC.
- [11] F. Carravetta, A. Germani, and M. Raimondi, “Polynomial filtering for linear discrete time non-Gaussian systems,” *SIAM Journal on Control and Optimization*, vol. 34, no. 5, pp. 1666–1690, 1996. [Online]. Available: citeseer.ist.psu.edu/carravetta96polynomial.html
- [12] —, “Polynomial filtering of discrete-time stochastic linear systems with multiplicative state noise,” in *IEEE Transactions on Automatic Control*, May 1997, pp. 1106–1126.
- [13] A. Germani, C. Manes, and P. Palumbo, “Polynomial Extended Kalman Filter,” in *IEEE Transactions on Automatic Control*, May 2005, pp. 2059–2064.
- [14] C. Manes, A. Martinelli, F. Martinelli, and P. Palumbo, “Mobile Robot Localization based on a Polynomial Approach,” in *Proceedings, IEEE Conference on Robotics and Automation, Roma, Italy*, April 2007, pp. 3539–3544.
- [15] K. Kowalski and W. Steeb, *Nonlinear Dynamical Systems and Carleman Linearization*. Singapore: World Scientific.
- [16] J. Guivant and E. Nebot, “Optimization of the simultaneous Localization and Map-building Algorithm for Real-Time Implementation,” *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 242–257, 2001.