On the Complexity and Consistency of UKF-based SLAM

Guoquan P. Huang, Anastasios I. Mourikis and Stergios I. Roumeliotis

Abstract— This paper addresses two key limitations of the unscented Kalman filter (UKF) when applied to the simultaneous localization and mapping (SLAM) problem: the cubic, in the number of states, computational complexity, and the inconsistency of the state estimates. In particular, we introduce a new sampling strategy that minimizes the linearization error and whose computational complexity is constant (i.e., independent of the size of the state vector). As a result, the overall computational complexity of UKF-based SLAM becomes of the same order as that of the extended Kalman filter (EKF) when applied to SLAM. Furthermore, we investigate the observability properties of the linear-regression-based model employed by the UKF, and propose a new algorithm, termed the Observability-Constrained (OC)-UKF, that improves the consistency of the state estimates. The superior performance of the OC-UKF compared to the standard UKF and its robustness to large linearization errors are validated by extensive simulations.

I. INTRODUCTION

For autonomous vehicles exploring unknown environments, the ability to perform simultaneous localization and mapping (SLAM) is essential. Among the algorithms developed thus far to solve the SLAM problem, the extended Kalman filter (EKF) arguably remains a popular choice and has been used in many SLAM applications. However, the EKF is also known to be vulnerable to linearization errors, which can cause poor performance or even divergence. A well-known problem with EKF-based SLAM is that the state estimates are typically *inconsistent*, which renders the estimator unreliable. In order to address the problems caused by linearization, the use of the unscented Kalman filter (UKF) [1], appears to be an appealing option. The UKF has been shown to generally perform better than the EKF in nonlinear estimation problems, and one would expect similar gains in the case of SLAM.

However, one of the main limitations of the original UKF algorithm [1] is its computational complexity, which is *cubic* in the size of the state vector. In the case of SLAM, where hundreds of landmarks are typically included in the state vector, this increased computational burden can preclude real-time operation. Moreover, when applied to SLAM, the performance gains of the UKF over the EKF are generally not overwhelming (e.g., [2], [3]). Most importantly, empirical evidence suggests that the UKF also results in inconsistent estimates in SLAM, even though its performance is better than the EKF in this respect.

G. P. Huang and S. I. Roumeliotis are with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA {ghuang|stergios}@cs.umn.edu

A. I. Mourikis is with the Department of Electrical Engineering, University of California, Riverside, CA 92521, USA mourikis@ee.ucr.edu

Our objective is to address the aforementioned limitations. In particular, the main contributions of this work are the following:

- We propose a new sampling strategy for UKF-based SLAM that has *constant* computational cost, regardless of the number of landmarks included in the state vector. This sampling strategy is provably optimal, in the sense that it minimizes the squared error between the nonlinear function and its linear approximation employed by the UKF. Using this strategy, the computational cost of UKF-based SLAM becomes *linear* during propagation and *quadratic* during update, which is in the same order as the EKF.

- We analytically examine the consistency of UKF-based SLAM, by studying the observability properties of the statistically-linearized system model, which is employed by the UKF. This analysis identifies a mismatch between the observability properties of this model and those of the underlying nonlinear system, which is a fundamental cause of inconsistency. Based on this theoretical analysis, we propose a novel UKF-based SLAM algorithm, termed Observability-Constrained UKF (OC-UKF). By imposing the appropriate observability constraints on the linear regression carried out by the UKF, the proposed OC-UKF ensures that its system model has observability properties similar to those of the actual, nonlinear, SLAM system. As a result, the OC-UKF outperforms the EKF and the standard UKF both in terms of accuracy, and in terms of consistency, as verified by simulation tests.

II. RELATED WORK

A number of previous approaches have applied the standard UKF algorithm to SLAM [2], [4], [5]. However, this algorithm, which involves computing the square root of the state covariance matrix at each time step, has computational complexity *cubic* in the number of landmarks, and thus is not suitable for real-time operation in large environments. To address this problem, Holmes *et al.* [3] recently proposed a square-root UKF (SRUKF) for monocular visual SLAM that has computational complexity *quadratic* both in the propagation and in the update phases. This approach offers a significant improvement in terms of computational complexity, at the cost of a significantly more complicated implementation. Additionally, as shown in [3], the algorithm is an order of magnitude slower than the standard EKF, due to the need to carry out expensive numerical computations.

Andrade-Cetto *et al.* [6] presented a "hybrid" EKF/UKF algorithm, where the EKF is employed in the update phase,

while the UKF is used during propagation for computing *only* the robot state and its covariance. The cross-correlation terms during propagation are handled in a fashion identical to the EKF. Even though this algorithm achieves computational complexity linear during propagation and quadratic during updates, the positive definiteness of the state covariance matrix cannot be guaranteed during propagation. Moreover, the use of the EKF for updates makes the approach vulnerable to large linearization errors. In contrast to the aforementioned approaches, the algorithm described in Section IV employs the unscented transformation *both* in the propagation and update phases, is simple to implement, and attains computational complexity *linear* during propagation, and *quadratic* during updates.

The issue of the *consistency* of UKF-based SLAM has not received considerable attention in the literature. In [2], [3] the consistency of the UKF was empirically examined, but, to the best of our knowledge, no theoretical results exist to date. On the other hand, the consistency of EKF-based SLAM has been studied in a number of publications [7]–[10]. In our recent work [11], [12], we have presented an analytical study of the issue, by focusing on the observability properties of the EKF's linearized system model. In this work, we extend this analysis to the case of UKF-based SLAM, and we analytically show that the implicit statistical linearization, performed by the UKF, results in a system model with "incorrect" observability properties, which is a fundamental cause of inconsistency.

III. LRKF AND UKF

In this section, we present the UKF in the context of the Linear-Regression Kalman Filter (LRKF). As shown in [13], the UKF is a special case of the LRKF, and therefore, it can be viewed as performing an implicit *statistical linearization* of the nonlinear propagation and update models. We here present the details of this linearization mechanism, which will be instrumental in the development of the quadratic-complexity UKF in Section IV.

A. Linear Regression

The LRKF approximates a nonlinear function $\mathbf{y} = \mathbf{g}(\mathbf{x})$ with a linear model $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b} + \mathbf{e}$, where $\mathbf{A}$ and $\mathbf{b}$ are a regression matrix and a regression vector, respectively, and $\mathbf{e}$ denotes the additional error term due to linearization. Once this linear approximation is computed, the LRKF proceeds by applying the regular Kalman filter equations. In computing the linear approximation of $\mathbf{g}(\mathbf{x})$, our goal is to minimize the average squared error of the approximation, which is defined as:

$$\min_{\mathbf{A},\mathbf{b}} \int_{-\infty}^{+\infty} [\mathbf{y} - (\mathbf{A}\mathbf{x} + \mathbf{b})]^T [\mathbf{y} - (\mathbf{A}\mathbf{x} + \mathbf{b})] p(\mathbf{x}) d\mathbf{x} \quad (1)$$

where $p(\mathbf{x})$ is the prior pdf of the state $\mathbf{x}$. Clearly, due to the nonlinear nature of $\mathbf{g}(\mathbf{x})$, computing the solution of this minimization problem in closed form is generally intractable. Therefore, in the LRKF, $r+1$ regression points and weights

$\{\mathcal{X}_i, w_i\}_{i=0}^r$ are selected, so that their sample mean and covariance equals the mean and covariance of the pdf, i.e.,[1]

$$\bar{\mathbf{x}} = \sum_{i=0}^r w_i \mathcal{X}_i = E\{\mathbf{x}\}$$

$$\bar{\mathbf{P}}_{\mathbf{xx}} = \sum_{i=0}^r w_i (\mathcal{X}_i - \bar{\mathbf{x}}) (\mathcal{X}_i - \bar{\mathbf{x}})^T = E\{(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T\}$$

Using the approximation $p(\mathbf{x}) \simeq \sum_{i=0}^r w_i \delta(\mathbf{x} - \mathcal{X}_i)$, (1) becomes:

$$\min_{\mathbf{A},\mathbf{b}} \sum_{i=0}^r w_i \underbrace{[\mathcal{Y}_i - (\mathbf{A}\mathcal{X}_i + \mathbf{b})]}_{\mathbf{e}_i}^T \underbrace{[\mathcal{Y}_i - (\mathbf{A}\mathcal{X}_i + \mathbf{b})]}_{\mathbf{e}_i} \quad (2)$$

where $\{\mathcal{Y}_i = \mathbf{g}(\mathcal{X}_i)\}_{i=0}^r$. This cost function is identical to that presented in [13], and the optimal solution for $\mathbf{A}$ and $\mathbf{b}$ is given by

$$\mathbf{A} = \bar{\mathbf{P}}_{\mathbf{yx}} \bar{\mathbf{P}}_{\mathbf{xx}}^{-1}, \qquad \mathbf{b} = \bar{\mathbf{y}} - \mathbf{A}\bar{\mathbf{x}} \quad (3)$$

where

$$\bar{\mathbf{y}} = \sum_{i=0}^r w_i \mathcal{Y}_i, \qquad \bar{\mathbf{P}}_{\mathbf{yx}} = \sum_{i=0}^r w_i (\mathcal{Y}_i - \bar{\mathbf{y}}) (\mathcal{X}_i - \bar{\mathbf{x}})^T \quad (4)$$

During recursive estimation, the LRKF employs the above statistical linearization procedure to approximate the nonlinear process and measurement models. It is important to note that, in this case, the regression matrices $\mathbf{A}$ and $\mathbf{b}$ serve as *inferred Jacobian* matrices, in a fashion similar to the EKF. The details are explained next.

B. Propagation

During propagation, the LRKF approximates the nonlinear process model by a linear function:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \boldsymbol{o}_k) \quad (5)$$

$$= \breve{\boldsymbol{\Phi}}_k \mathbf{x}_k + \breve{\mathbf{G}}_k \boldsymbol{o}_k + \mathbf{b}_k + \mathbf{e}_k \quad (6)$$

$$= \begin{bmatrix} \breve{\boldsymbol{\Phi}}_k & \breve{\mathbf{G}}_k \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{o}_k \end{bmatrix} + \mathbf{b}_k + \mathbf{e}_k \quad (7)$$

where $\mathbf{x}_\ell$ is the system state vector at time ℓ, $\ell = k, k+1$, $\boldsymbol{o}_k = \boldsymbol{o}_{m_k} - \mathbf{w}_k$ is the control input (e.g., odometry), $\boldsymbol{o}_{m_k}$ is the corresponding measurement, and $\mathbf{w}_k$ is the odometry noise vector, assumed to be zero-mean, white, and Gaussian, with covariance matrix $\mathbf{Q}_k$. The matrices $\breve{\boldsymbol{\Phi}}_k$ and $\breve{\mathbf{G}}_k$ can be viewed as inferred Jacobians, in an analogy to the linearized approximation of the EKF.

In the LRKF propagation step, $r + 1$ sample points $\{\mathcal{X}_i(k)\}_{i=0}^r$ are selected based on the *augmented* vector that comprises the filter state and the control input. The

---

[1] Throughout this paper, $\bar{\boldsymbol{\alpha}}$ and $\bar{\mathbf{P}}_{\boldsymbol{\alpha\alpha}}$ denote the sample mean and covariance of regression points $\mathcal{A}_i$, drawn from the pdf of the variable $\boldsymbol{\alpha}$. $\bar{\mathbf{P}}_{\boldsymbol{\alpha\beta}}$ denotes the sample cross-correlation between sets of samples $\mathcal{A}_i$ and $\mathcal{B}_i$, drawn from the pdf of the variables $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively. $\hat{x}$ is used to denote the estimate of a random variable x, and $\tilde{x} = x - \hat{x}$ is the error in this estimate. Finally, the subscript $\ell|j$ refers to the estimate of a quantity at time-step ℓ, after all measurements up to time-step j have been processed.

sample mean and sample covariance, respectively, of the set $\{\mathcal{X}_i(k)\}_{i=0}^r$ is:

$$\bar{\mathbf{x}}_{k|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \mathbf{o}_{m_k} \end{bmatrix}, \qquad \bar{\mathbf{P}}_{\mathbf{xx}_{k|k}} = \begin{bmatrix} \mathbf{P}_{k|k} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix} \qquad (8)$$

Then the LRKF produces the regression points, $\{\mathcal{Y}_i(k) = \mathcal{X}_i(k+1)\}_{i=0}^r$, by passing the sample points through the non-linear process function (5). The sample mean and covariance of the $\mathcal{Y}_i$ points is used as the mean and covariance matrix of the propagated state estimates. Moreover, the inferred Jacobian matrices $\breve{\boldsymbol{\Phi}}_k$ and $\breve{\mathbf{G}}_k$ can be computed from the expression (cf. [13], [14]):

$$\begin{bmatrix} \breve{\boldsymbol{\Phi}}_k & \breve{\mathbf{G}}_k \end{bmatrix} = \bar{\mathbf{P}}_{\mathbf{yx}_{k|k}} \bar{\mathbf{P}}_{\mathbf{xx}_{k|k}}^{-1} \qquad (9)$$

C. Update

During update, the LRKF employs statistical linearization in order to approximate the nonlinear measurement function:

$$\mathbf{z}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1} \qquad (10)$$
$$= \breve{\mathbf{H}}_{k+1}\mathbf{x}_{k+1} + \mathbf{b}'_{k+1} + \mathbf{v}_{k+1} + \mathbf{e}'_{k+1} \qquad (11)$$

where $\mathbf{z}_{k+1}$ is the measurement and $\mathbf{v}_{k+1}$ is the measurement noise vector, assumed to be zero-mean, white, and Gaussian, with covariance matrix $\mathbf{R}_{k+1}$. A set of $r+1$ points with sample mean and covariance equal to $\hat{\mathbf{x}}_{k+1|k}$ and $\mathbf{P}_{k+1|k}$, respectively, are selected, and then passed through the non-linear measurement function (10), to obtain $r+1$ regression points: $\{\mathcal{Z}_i(k+1) = \mathbf{h}(\mathcal{X}_i(k+1))\}_{i=0}^r$. The regression matrix (i.e., the inferred Jacobian) $\breve{\mathbf{H}}_{k+1}$ can then be obtained as (cf. (3)):

$$\breve{\mathbf{H}}_{k+1} = \bar{\mathbf{P}}_{\mathbf{zx}_{k+1|k}} \mathbf{P}_{k+1|k}^{-1} \qquad (12)$$

Subsequently, the state and covariance are updated using the filter update equations:

$$\mathbf{S}_{k+1} = \bar{\mathbf{P}}_{\mathbf{zz}_{k+1|k}} + \mathbf{R}_{k+1} \qquad (13)$$
$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \breve{\mathbf{H}}_{k+1}^T \mathbf{S}_{k+1}^{-1} \qquad (14)$$
$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}(\mathbf{z}_{k+1} - \bar{\mathbf{z}}_{k+1}) \qquad (15)$$
$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}_{k+1}^T \qquad (16)$$

D. UKF

As argued in [13], the UKF is a special case of the LRKF. It deterministically chooses $r+1 = 2n+1$ so-called sigma points $\mathcal{X}_i$ in the n-dimensional state space, with weights w_i, $i = 1, \ldots, n$, according to the equations:

$$\mathcal{X}_0(\ell|k) = \hat{\mathbf{x}}_{\ell|k} \, , \; w_0 = \frac{2\kappa}{2(n+\kappa)} \qquad (17)$$

$$\mathcal{X}_i(\ell|k) = \hat{\mathbf{x}}_{\ell|k} + \left(\sqrt{(n+\kappa)\mathbf{P}_{\ell|k}}\right)_i \, , \; w_i = \frac{1}{2(n+\kappa)}$$

$$\mathcal{X}_{i+n}(\ell|k) = \hat{\mathbf{x}}_{\ell|k} - \left(\sqrt{(n+\kappa)\mathbf{P}_{\ell|k}}\right)_i \, , \; w_{i+n} = \frac{1}{2(n+\kappa)}$$

where $\left(\sqrt{(n+\kappa)\mathbf{P}_{\ell|k}}\right)_i$ is the ith column of $\sqrt{(n+\kappa)\mathbf{P}_{\ell|k}}$, and $\ell = k$ or $k+1$. κ is a degree of freedom in the choice of the sigma points $\mathcal{X}_i$, usually chosen so that $n+\kappa = 3$. This set of sigma points captures the moments of the underlying distribution up to the third-order for the Gaussian case.

IV. Quadratic-Complexity UKF-based SLAM

In this section, we show how the computational cost of the UKF, when applied to the SLAM problem, can be optimized. In particular, in this paper we focus on 2-D SLAM, in which case the state vector $\mathbf{x}_k$ comprises the robot pose and landmark positions, i.e.,

$$\mathbf{x}_k^T = \begin{bmatrix} \mathbf{x}_{R_k}^T & \mathbf{p}_{L_1}^T & \cdots & \mathbf{p}_{L_M}^T \end{bmatrix} \qquad (18)$$

where $\mathbf{x}_{R_k}^T = \begin{bmatrix} \mathbf{p}_{R_k}^T & \phi_{R_k} \end{bmatrix}$ denotes the robot pose (position and orientation), and $\mathbf{p}_{L_i}$, $i = 1 \ldots M$, are the positions of the M landmarks.

In the UKF algorithm presented in the preceding section, the main bottleneck is the computation of the square root matrix of the covariance, which has complexity $\mathcal{O}(M^3)$. Clearly, in a scenario where hundreds of landmarks are included in the state vector, carrying out this operation during each propagation and update step would incur an unacceptable computational burden. To address this problem, we here propose a new sampling scheme for the UKF, which has computational cost $\mathcal{O}(1)$, during both propagation and update. The derivation of this sampling scheme is based on the observation that, during SLAM, only a *small subset* of the filter states appear in the nonlinear propagation and measurement models. In particular, during propagation only the robot state changes, while each measurement involves only the robot pose and one landmark. To take advantage of this important property, we employ the following lemma:

Lemma 4.1: Let the vector $\mathbf{x}$ be partitioned as $\mathbf{x}^T = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T \end{bmatrix}$, and let $\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}_1)$ (i.e., only the state entries of $\mathbf{x}_1$ appear in $\mathbf{g}(\mathbf{x})$). Moreover, let the linear regression matrix $\mathbf{A}$ be partitioned as $\mathbf{A} = [\mathbf{A}_1 \quad \mathbf{A}_2]$, i.e.,

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b} + \mathbf{e} = \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 + \mathbf{b} + \mathbf{e} \qquad (19)$$

Then the optimal solution to (2) is:

$$\mathbf{A}_1 = \bar{\mathbf{P}}_{\mathbf{yx}_1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}, \quad \mathbf{A}_2 = \mathbf{0}, \quad \mathbf{b} = \bar{\mathbf{y}} - \mathbf{A}_1\hat{\mathbf{x}}_1 \qquad (20)$$

Proof: The linearization error is written as

$$\mathbf{e}(\mathbf{x}) = \mathbf{y} - (\mathbf{A}\mathbf{x} + \mathbf{b}) = \mathbf{y} - \mathbf{A}_1\mathbf{x}_1 - \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}$$

Substituting in the expression for the expected value of the quadratic linearization error, we have:

$$c = \int_{-\infty}^{+\infty} \mathbf{e}(\mathbf{x})^T\mathbf{e}(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int_{-\infty}^{+\infty} \mathbf{e}^*(\mathbf{x}_1)^T\mathbf{e}^*(\mathbf{x}_1)p(\mathbf{x})d\mathbf{x}$$
$$+ \int_{-\infty}^{+\infty} \mathbf{x}_2^T\mathbf{A}_2^T\mathbf{A}_2\mathbf{x}_2p(\mathbf{x})d\mathbf{x} - 2\int_{-\infty}^{+\infty} \mathbf{e}^*(\mathbf{x}_1)^T\mathbf{A}_2\mathbf{x}_2p(\mathbf{x})d\mathbf{x} \qquad (21)$$

with

$$\mathbf{e}^*(\mathbf{x}_1) = \mathbf{y} - \mathbf{A}_1\mathbf{x}_1 - \mathbf{b} \qquad (22)$$
$$p(\mathbf{x}) = \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P}) \qquad (23)$$

where

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathbf{x}_1\mathbf{x}_1} & \mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} \\ \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1} & \mathbf{P}_{\mathbf{x}_2\mathbf{x}_2} \end{bmatrix} \qquad (24)$$

Using the identity $p(\mathbf{x}) = p(\mathbf{x}_2|\mathbf{x}_1)p(\mathbf{x}_1)$ and the approximation $p(\mathbf{x}_1) \simeq \sum_i w_i \delta(\mathbf{x}_1 - \mathcal{X}_{1_i})$, the cost function (21) becomes [14]:

$$c \simeq \sum_i w_i (\mathcal{Y}_i - \mathbf{A}_1 \mathcal{X}_{1_i} - \mathbf{b})^T (\mathcal{Y}_i - \mathbf{A}_1 \mathcal{X}_{1_i} - \mathbf{b}) +$$
$$\mathrm{tr}\left[\mathbf{A}_2 (\mathbf{P}_{\mathbf{x}_2 \mathbf{x}_2} + \hat{\mathbf{x}}_2 \hat{\mathbf{x}}_2^T) \mathbf{A}_2^T\right] -$$
$$2 \sum_i w_i (\mathcal{Y}_i - \mathbf{A}_1 \mathcal{X}_{1_i} - \mathbf{b})^T \mathbf{A}_2 \hat{\mathcal{X}}_{2_i} \quad (25)$$

where $\hat{\mathcal{X}}_{2_i} := \hat{\mathbf{x}}_2 + \mathbf{P}_{\mathbf{x}_2 \mathbf{x}_1} \mathbf{P}_{\mathbf{x}_1 \mathbf{x}_1}^{-1} (\mathcal{X}_{1_i} - \hat{\mathbf{x}}_1)$, $\mathcal{X}_{1_i}$ are sample points drawn from the distribution $p(\mathbf{x}_1) = \mathcal{N}(\hat{\mathbf{x}}_1, \mathbf{P}_{\mathbf{x}_1 \mathbf{x}_1})$, and $\mathcal{Y}_i = \mathbf{g}(\mathcal{X}_{1_i})$.

Setting the derivatives of the cost function (25) with respect to $\mathbf{b}$, $\mathbf{A}_1$, and $\mathbf{A}_2$ to zero, we obtain (20). ∎

This result means that, in order to minimize the expected error of the statistical linearization (cf. (2)), it suffices to draw sample points from the pdf of $\mathbf{x}_1$. In SLAM, the number of states participating in the nonlinear propagation and measurement models is constant, and thus we can reduce the cost of UKF sampling to $\mathcal{O}(1)$, by applying the unscented transformation only to the pertinent state entries, instead of sampling over the full state. Compared to EKF-SLAM, the proposed UKF-SLAM only incurs a small computational overhead (for computing the square roots of constant-size matrices), and has the same complexity. The details of the new sampling strategy are presented in the following.

A. Propagation

During propagation, only the robot pose and the control input (odometry) participate in the process model. Therefore, we are able to reduce the computational complexity by applying the unscented transformation only to the part of the state comprising the robot pose and the control input, instead of the full state vector. Specifically, we draw the sigma points based on the vector with mean and covariance:

$$\bar{\mathbf{x}}_{k|k} = \begin{bmatrix} \hat{\mathbf{x}}_{R_{k|k}} \\ \boldsymbol{o}_{m_k} \end{bmatrix}, \quad \bar{\mathbf{P}}_{\mathbf{xx}_{k|k}} = \begin{bmatrix} \mathbf{P}_{RR_{k|k}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix} \quad (26)$$

where $\mathbf{P}_{RR_{k|k}}$ is the covariance matrix corresponding to the robot pose, obtained by partitioning the state covariance matrix as $\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{RR_{k|k}} & \mathbf{P}_{RL_{k|k}} \\ \mathbf{P}_{RL_{k|k}}^T & \mathbf{P}_{LL_{k|k}} \end{bmatrix}$. Note that the vector $\bar{\mathbf{x}}_{k|k}$ is of dimension 5 (assuming that the odometry measurement $\boldsymbol{o}_{m_k}$ is 2-dimensional), and thus the computational cost of computing the sigma points is very small. Subsequently, we transform each of the sigma points using the process model (5), to obtain samples of the propagated robot pose, $\mathcal{Y}_i = \mathcal{X}_{R_i}(k+1|k)$, $i = 0, \ldots, 10$. This enables us to compute the mean and covariance, of the propagated robot state, in the same way as in the standard LRKF/UKF (cf. Section III-B). Moreover, we can evaluate the *inferred* propagation Jacobian:

$$\mathbf{A} = \bar{\mathbf{P}}_{\mathbf{yx}_{k|k}} \bar{\mathbf{P}}_{\mathbf{xx}_{k|k}}^{-1} = \begin{bmatrix} \breve{\boldsymbol{\Phi}}_{R_k} & \breve{\mathbf{G}}_{R_k} \end{bmatrix} \quad (27)$$

In order to compute the cross-correlation between the propagated robot state and the landmarks, we note that:

$$\mathbf{P}_{RL_{k+1|k}} = E\left\{ \tilde{\mathbf{x}}_{R_{k+1|k}} \tilde{\mathbf{x}}_{L_{k|k}}^T \right\}$$
$$= E\left\{ \left(\breve{\boldsymbol{\Phi}}_{R_k} \tilde{\mathbf{x}}_{R_{k|k}} + \breve{\mathbf{G}}_{R_k} \mathbf{w}_k + \mathbf{e}_k \right) \tilde{\mathbf{x}}_{L_{k|k}}^T \right\}$$
$$= \breve{\boldsymbol{\Phi}}_{R_k} \mathbf{P}_{RL_{k|k}} \quad (28)$$

Thus the propagated SLAM covariance matrix is given by:

$$\mathbf{P}_{k+1|k} = \begin{bmatrix} \bar{\mathbf{P}}_{\mathbf{yy}_{k|k}} & \breve{\boldsymbol{\Phi}}_{R_k} \mathbf{P}_{RL_{k|k}} \\ \mathbf{P}_{RL_{k|k}}^T \breve{\boldsymbol{\Phi}}_{R_k}^T & \mathbf{P}_{LL_{k|k}} \end{bmatrix} \quad (29)$$

Clearly, the computational cost of propagation is *linear* in the size of the state vector.

B. Update

Any measurement used for updating involves only the robot pose and the position of the observed landmark. Therefore, we can apply the unscented transformation only to this subset of the states, to reduce the computational cost. In particular, assume the jth landmark is observed at time step $k + 1$. Then, the set of sigma points $\{\mathcal{X}_i\}_{i=0}^{10}$ is drawn with mean and covariance:

$$\bar{\mathbf{x}}_{k+1|k} = \begin{bmatrix} \hat{\mathbf{x}}_{R_{k+1|k}} \\ \hat{\mathbf{x}}_{L_{j,k+1|k}} \end{bmatrix}, \quad \bar{\mathbf{P}}_{\mathbf{xx}_{k+1|k}} = \begin{bmatrix} \mathbf{P}_{RR_{k+1|k}} & \mathbf{P}_{RL_{j,k+1|k}} \\ \mathbf{P}_{L_j R_{k+1|k}} & \mathbf{P}_{L_j L_{j,k+1|k}} \end{bmatrix} \quad (30)$$

Note that this process involves a matrix with *constant* size, regardless of the number of landmarks in the state vector. Once the set of sigma points is generated, the procedure of the LRKF update (cf. Section III-C) is applied, to obtain the *inferred* measurement Jacobian:

$$\mathbf{A} = \bar{\mathbf{P}}_{\mathbf{zx}_{k+1|k}} \bar{\mathbf{P}}_{\mathbf{xx}_{k+1|k}}^{-1} = \begin{bmatrix} \breve{\mathbf{H}}_{R_{k+1}} & \breve{\mathbf{H}}_{L_{j,k+1}} \end{bmatrix} \quad (31)$$

where the submatrix $\breve{\mathbf{H}}_{R_{k+1}}$ corresponds to the robot pose, while $\breve{\mathbf{H}}_{L_{j,k+1}}$ corresponds to the jth landmark. To construct the inferred measurement Jacobian for the entire state vector, we note that all terms corresponding to the landmarks not currently observed are zero. Thus, the entire *inferred* measurement Jacobian is obtained as:

$$\breve{\mathbf{H}}_{k+1} = \begin{bmatrix} \breve{\mathbf{H}}_{R_{k+1}} & \mathbf{0} & \cdots & \mathbf{0} & \breve{\mathbf{H}}_{L_{j,k+1}} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \quad (32)$$

Once this matrix is available, equations (13)-(16) are applied to update the state and covariance in the UKF. It is important to point out that the computational complexity of these equations is determined by the covariance update equation, and is quadratic in the number of landmarks, similarly to the EKF.

V. SLAM Observability Analysis

As discussed in Section III, the UKF (as a special case of the LRKF) relies on a linear approximation of the nonlinear system, in order to carry out the recursive estimation. The system model employed by the UKF is given by equations (7) and (11). Thus, the properties of this system model are important for determining the performance of the estimator. As shown in [11] for the case of the EKF, when a linearized system model is employed for estimation, the *observability properties* of this model are crucial in

determining filter consistency. The observability properties of the UKF's linear-regression-based model can be studied by examining the observability matrix for the time interval between time-steps k_o and $k_o + k$, defined as:

$$\mathbf{M} = \begin{bmatrix} \breve{\mathbf{H}}_{k_o} \\ \breve{\mathbf{H}}_{k_o+1}\breve{\mathbf{\Phi}}_{k_o} \\ \vdots \\ \breve{\mathbf{H}}_{k_o+k}\breve{\mathbf{\Phi}}_{k_o+k-1}\cdots\breve{\mathbf{\Phi}}_{k_o} \end{bmatrix} \quad (33)$$

We see that $\mathbf{M}$ is a function of the regression matrices used in the UKF, and thus the structure of these matrices will in turn determine the observability properties of the system model. Ideally, we would require these properties to match those of the underlying *nonlinear* SLAM system, which has 3 unobservable degrees of freedom corresponding to the global position and orientation of the state vector. In [11], it was shown that this would be the case, if we could create an "oracle" or "ideal" EKF, where linearization is carried out using Jacobians evaluated at the *true* values of the state. Specifically, if instead of the regression matrices $\breve{\mathbf{H}}_k$ and $\breve{\mathbf{\Phi}}_k$ we employ the "ideal" Jacobians:

$$\mathbf{H}_k = \nabla_{\mathbf{x}_k}\mathbf{h}\Big|_{\mathbf{x}_{k_{\text{true}}}}, \qquad \mathbf{\Phi}_k = \nabla_{\mathbf{x}_k}\mathbf{f}\Big|_{\mathbf{x}_{k_{\text{true}}}} \quad (34)$$

then the observability matrix, in the case where a *single* landmark is included in the state vector, is equal to [11]:

$$\mathbf{M}_{\text{ideal}} = \mathbf{D} \times \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_L - \mathbf{p}_{R_{k_o}}) & \mathbf{I}_2 \\ -\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_L - \mathbf{p}_{R_{k_o}}) & \mathbf{I}_2 \\ \vdots & \vdots & \vdots \\ -\mathbf{I}_2 & -\mathbf{J}(\mathbf{p}_L - \mathbf{p}_{R_{k_o}}) & \mathbf{I}_2 \end{bmatrix} \quad (35)$$

where $\mathbf{D}$ is a block-diagonal matrix with full row-rank, $\mathbf{I}_2$ is the 2×2 identity matrix, and $\mathbf{J} \triangleq \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. The rank of this observability matrix is two, which implies that the system model of the ideal EKF has three unobservable directions. A basis for the unobservable subspace (i.e., a basis for the right nullspace of $\mathbf{M}_{\text{ideal}}$) is:

$$\mathcal{N}(\mathbf{M}_{\text{ideal}}) = \operatorname*{span}_{\text{col.}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\mathbf{p}_{R_{k_o}} \\ \mathbf{0}_{1\times 2} & 1 \\ \mathbf{I}_2 & \mathbf{J}\mathbf{p}_L \end{bmatrix} \triangleq \operatorname*{span}_{\text{col.}} \begin{bmatrix} \mathbf{n}_1 & \mathbf{n}_2 & \mathbf{n}_3 \end{bmatrix} \quad (36)$$

We observe that the vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ correspond to a "shifting" of the $x-y$ plane, while the vector $\mathbf{n}_3$ corresponds to a rotation of the $x - y$ plane. These are precisely the unobservable directions of the underlying, nonlinear SLAM system, which shows that the ideal EKF offers a good approximation to the nonlinear system (from the perspective of observability). We point out that even though the above analysis only addresses the case of a single landmark, analogous results are obtained for the general case where $M > 1$ landmarks are included in the state vector [14].

Since the UKF employs the linearization of (7) and (11) to approximate the nonlinear SLAM system, we would require that the corresponding system model has observability properties similar to those of the nonlinear system, and consequently, identical to those of the ideal EKF. A necessary and sufficient condition for this to occur is that the vectors $\mathbf{n}_i$, $i = 1, 2, 3$, lie within the nullspace of *each* of the block rows of the observability matrix, i.e.,

$$\mathbf{n}_i \in \mathcal{N}(\breve{\mathbf{H}}_{k_o}), \quad \dots \quad, \quad \mathbf{n}_i \in \mathcal{N}(\breve{\mathbf{H}}_{k_o+k}\breve{\mathbf{\Phi}}_{k_o+k-1}\cdots\breve{\mathbf{\Phi}}_{k_o})$$

for $i = 1, 2, 3$. However, this is generally *not* the case. In fact, when numerically computing the dimension of the nullspace of the observability matrix, we find that it is *less* than three. This implies that the UKF obtains "spurious" information, in directions of the state space where no information is available. This, in turn will lead to an unjustified reduction of the covariance matrix of the estimates, thus causing inconsistency. This fact, which has been generally overlooked in the literature, can cause a significant degradation in filter performance, as shown in the simulation results of Section VII.

VI. Observability-Constrained UKF SLAM

In this section, we propose a novel UKF that employs a system model with observability properties similar to those of the actual nonlinear SLAM system. In our previous work [11], we proposed the conjecture that the observability properties of the linearized error-state model in EKF-SLAM play a fundamental role in determining consistency. Motivated by this conjecture, we derived the First-Estimate Jacobian (FEJ)-EKF, which always employs the first state estimates in computing the Jacobians, thus achieving the desired observability properties. This filter outperforms the standard EKF in SLAM, even though it uses "older" (and thus less accurate) state estimates for Jacobian evaluation.

We here propose an analogous approach, within the UKF framework. Specifically, we construct the "inferred" Jacobians of the UKF in such a way that the resulting system model has an unobservable subspace of dimension three. Even though these regression matrices do *not* generally minimize the expected squared error of linearization (cf. (2)), we have verified in simulation that the resulting filter, termed Observability Constrained (OC)-UKF, outperforms the standard UKF in terms of both accuracy and consistency (cf. Section VII). This further supports our conjecture about the importance of the observability properties of the filter's system model in determining consistency.

To maintain clarity, we here present the OC-UKF for the case where a single landmark is included in the SLAM state vector. The more general case is handled in the same way, and the interested reader is referred to [14] for details.

The propagation phase of the OC-UKF is identical to that of the standard UKF. The difference arises in the update phase, where, instead of employing the unconstrained minimization of (2) for computing the regression matrix $\mathbf{A}$, we employ *constrained* optimization, by enforcing the desired observability properties. In particular, if a landmark

was first observed at time-step k_o, we require that

$$\breve{\mathbf{H}}_{k_o}\mathbf{N} = \mathbf{0} \tag{37}$$

$$\breve{\mathbf{H}}_{k_o+k}\breve{\boldsymbol{\Phi}}_{k_o+k-1}\cdots\breve{\boldsymbol{\Phi}}_{k_o}\mathbf{N} = \mathbf{0}, \; k > 0 \tag{38}$$

In this expression, $\mathbf{N}$ is a 5×3 matrix, whose columns span the desired nullspace. These constraints ensure that all the block rows of the observability matrix $\mathbf{M}$ (cf. (33)) have the *same* nullspace, which coincides with the *unobservable subspace* of the filter's system model. It is interesting to note that, in this formulation, this subspace is a design choice, since we can freely choose $\mathbf{N}$. For instance, we can choose a matrix whose columns span the nullspace of the *first* inferred Jacobian, $\breve{\mathbf{H}}_{k_o}$ (which can be computed via the SVD of $\breve{\mathbf{H}}_{k_o}$), or we can explicitly require the nullspace to be of the same structure as that of the ideal EKF, by choosing:

$$\mathbf{N} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{R_{k_o|k_o}} \\ \mathbf{0}_{1\times 2} & 1 \\ \mathbf{I}_2 & \mathbf{J}\hat{\mathbf{p}}_{L_{k_o|k_o}} \end{bmatrix} \tag{39}$$

Note that, since the true value of the state is unknown in (39), we employ the state estimates at time-step k_o.

The regression matrix $\mathbf{A}$ at each update step (cf. (31)) is obtained by solving the constrained optimization problem:

$$\min_{\mathbf{A},\mathbf{b}} \quad \sum_{i=0}^{10} w_i \mathbf{e}_i^T \mathbf{e}_i \tag{40}$$

$$\text{s.t.} \quad \mathbf{A}\breve{\boldsymbol{\Phi}}_{k_o+k-1}\cdots\breve{\boldsymbol{\Phi}}_{k_o}\mathbf{N} = \mathbf{0} \tag{41}$$

where $\breve{\boldsymbol{\Phi}}_\ell = \begin{bmatrix} \breve{\boldsymbol{\Phi}}_{R_\ell} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix}$ denotes the regression matrix obtained from propagation (cf. (27)), corresponding to the system whose state vector comprises the robot pose and the landmark position. The sigma points used in the above minimization problem are computed by the procedure described in Section IV-B.

We now derive a solution to the constrained optimization problem (40) in *closed form*. First, we define $\mathbf{U} \triangleq \breve{\boldsymbol{\Phi}}_{k_o+k-1}\cdots\breve{\boldsymbol{\Phi}}_{k_o}\mathbf{N}$, and using this notation, we write the equality constraint on $\mathbf{A}$ as $\mathbf{A}\mathbf{U} = \mathbf{0}$. This equation states that the rows of $\mathbf{A}$ lie in the left nullspace of the 5×3 matrix $\mathbf{U}$. Therefore, if $\mathbf{L}$ is a 2×5 matrix whose rows span this nullspace, we can write $\mathbf{A}$ as:

$$\mathbf{A} = \mathbf{B}\mathbf{L} \tag{42}$$

where $\mathbf{B}$ is the unknow 2×2 matrix that we seek to compute. We note that there are several possible ways of computing an appropriate matrix $\mathbf{L}$, whose rows lie in the nullspace of $\mathbf{U}$. For instance, such a matrix is given, in closed form, by the expression:

$$\mathbf{L} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_{2\times 3} \end{bmatrix}\left(\mathbf{I}_5 - \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\right) \tag{43}$$

Substituting (42) in the original problem formulation (cf. (2), (40)), we obtain:

$$\min_{\mathbf{B},\mathbf{b}} \quad \sum_{i=0}^{10} w_i \left(\mathcal{Z}_i - (\mathbf{B}\mathcal{L}_i + \mathbf{b})\right)^T \left(\mathcal{Z}_i - (\mathbf{B}\mathcal{L}_i + \mathbf{b})\right) \tag{44}$$

where we have defined $\mathcal{L}_i = \mathbf{L}\mathcal{X}_i$, $i = 0,\ldots,10$. This is an unconstrained minimization problem with design variables $\mathbf{B}$ and $\mathbf{b}$, and has exactly the same structure as that in (2). By analogy, we thus see that the optimal solution for $\mathbf{B}$ is

$$\mathbf{B} = \bar{\mathbf{P}}_{\mathbf{z}\ell}\mathbf{P}_{\ell\ell}^{-1} \tag{45}$$

where

$$\bar{\mathbf{P}}_{\mathbf{z}\ell} = \sum_{i=0}^{10} w_i(\mathcal{Z}_i - \bar{\mathbf{z}})(\mathbf{L}\mathcal{X}_i - \mathbf{L}\bar{\mathbf{x}})^T = \bar{\mathbf{P}}_{\mathbf{z}\mathbf{x}}\mathbf{L}^T$$

$$\mathbf{P}_{\ell\ell} = \sum_{i=0}^{10} w_i(\mathbf{L}\mathcal{X}_i - \mathbf{L}\bar{\mathbf{x}})(\mathbf{L}\mathcal{X}_i - \mathbf{L}\bar{\mathbf{x}})^T = \mathbf{L}\bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}}\mathbf{L}^T$$

By combining these two results with those of (45) and (42), we obtain the optimal value of $\mathbf{A}$ as:

$$\mathbf{A} = \begin{bmatrix} \breve{\mathbf{H}}_{R_{k_o+k}} & \breve{\mathbf{H}}_{L_{k_o+k}} \end{bmatrix} = \bar{\mathbf{P}}_{\mathbf{z}\mathbf{x}}\mathbf{L}^T \left(\mathbf{L}\bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}}\mathbf{L}^T\right)^{-1}\mathbf{L} \tag{46}$$

After constructing the inferred measurement Jacobian matrix $\breve{\mathbf{H}}_{k_o+k}$ from the regression matrix $\mathbf{A}$ (cf. (32)), we are able to update the state and its covariance (cf. (13)-(16)). We point out that in the general case, where multiple landmarks are observed, the above process is repeated for each of the landmarks observed at the current time step. The maximal dimension of all the matrices involved is 5 (cf. (30)), and thus computing the regression matrix $\mathbf{A}$ incurs only a *constant* computational overhead, regardless of the number of landmarks in the state. As a result, the overall computational cost of the update step remains quadratic, and is dominated by the cost of updating the covariance (cf. (16)). The proposed OC-UKF SLAM is summarized in Algorithm 1.

Algorithm 1 Observability-Constrained (OC)-UKF SLAM

Require: Initial state estimate and covariance
1: **loop**
2:    Propagation:
3:      Determine sigma points by (17) with mean and covariance (26).
4:      Produce regression points by passing the sigma points through (5) and compute the propagated state estimate by sample mean.
5:      Compute regression matrices via (27).
6:      Compute propagated covariance via (29).

7:    Update:
8:      Determine sigma points by (17) with mean and covariance (30).
9:      Produce regression points by passing the sigma points through (10).
10:      Compute regression matrix via (46) and (32).
11:      Update state and covariance via (13)-(16).
12: **end loop**

VII. SIMULATION RESULTS

A series of Monte-Carlo comparison studies were conducted under various conditions, in order to demonstrate

the capability of the OC-UKF to improve the consistency of UKF-based SLAM. The metrics used to evaluate filter performance are: (i) the RMS error, and (ii) the average normalized (state) estimation error squared (NEES) [15]. Specifically, for the landmarks we compute the average RMS errors and average NEES by averaging the squared errors and the NEES, respectively, over all Monte Carlo runs, all landmarks, and all time steps. On the other hand, for the robot pose we compute these error metrics by averaging over all Monte Carlo runs for each time step (cf. [14] for a more detailed description). By studying both the RMS errors and NEES, we obtain a comprehensive picture of the estimators' performance.

In the simulation tests presented in this section, a robot with a simple 3-wheel (2 active and 1 caster) kinematic model moves on a planar surface, at a constant velocity of $v = 0.25$ m/sec. The two active wheels are equipped with encoders, which measure their revolutions and provide measurements of velocity with standard deviation equal to $\sigma = 2\%v$ for each wheel. These measurements are used to obtain linear velocity measurements for the robot, with standard deviation equal to $\sigma_v = \frac{\sigma}{\sqrt{2}}$, and rotational velocity measurements with standard deviation $\sigma_\omega = 2\sqrt{2}\sigma$. The robot records distance and bearing measurements to landmarks that lie within its sensing range of 5 m. The standard deviation of the distance measurement noise is equal to 10% of the robot-to-landmark distance, while the standard deviation of the bearing measurement noise is set to 10 deg. It should be pointed out that the sensor-noise levels selected for the simulations are larger than what is typically encountered in practice. This was done on purpose, so as to make the effects of inconsistency more apparent (larger noise levels lead to larger estimation errors, and thus less accurate linearization). We note that the initialization of the landmarks is performed by use of the unscented transformation, as detailed in [14].

For the results shown here, a SLAM scenario with multiple loop closures was considered, where during each run, the robot executes 8 loops on a circular trajectory, and observes 20 landmarks in total. The reported results are averages over 50 Monte Carlo simulations. During the test, five filters process the same data, to ensure a fair comparison[2]. These are: (i) the ideal EKF, (ii) the standard EKF, (iii) the FEJ-EKF [11], (iv) the standard UKF, and (v) the OC-UKF.

The comparative results for all filters are presented in Fig. 1 and Table I. Specifically, Fig. 1(a) and Fig. 1(b) show the average NEES and RMS errors for the robot pose, respectively. On the other hand, Table I presents the average values of all relevant performance metrics for the landmarks and robot.

Several interesting conclusions can be drawn from these results. First, it becomes clear that the performance of

[2]In [11], the FEJ-EKF was shown to perform better, in terms of accuracy and consistency, than the robocentric mapping algorithm [16], which aims at improving the consistency of SLAM by expressing the landmarks in a robot-relative frame. Therefore, in this paper we omit the comparison between the proposed OC-UKF and robocentric mapping.

Ideal EKF	Std EKF	FEJ-EKF	Std UKF	OC-UKF
Robot Position Err. RMS (m)				
0.4386	0.6320	0.4712	0.6009	0.4491
Robot Heading Err. RMS (rad)				
0.0415	0.0546	0.0445	0.0523	0.0424
Robot Pose NEES				
3.0922	6.4944	4.1061	6.0324	3.4663
Landmark Position Err. RMS (m)				
0.4506	0.6743	0.4842	0.6373	0.4627
Landmark Position NEES				
1.9922	5.4125	3.3575	4.9038	2.3611

TABLE I
ROBOT POSE AND LANDMARK POSITION ESTIMATION PERFORMANCE

the OC-UKF is *very close* to that of the ideal EKF, and substantially better than both the standard EKF and the standard UKF, in terms of RMS errors, as well as in terms of NEES. The observed performance gains indicate that the observability properties of the linear-regression-based system model employed in the UKF play a key role in determining the filter consistency. If these properties differ from those of the underlying nonlinear system, consistency cannot be guaranteed. Moreover, the results shown here agree with those obtained in the case of the EKF in [11]. This further validates the claim that the observability properties of a filter's system model significantly impact the filter's performance.

A second observation is that, in this large sensor-noise setting, the OC-UKF also outperforms the FEJ-EKF [11], both in terms of accuracy and in terms of consistency. On the other hand, in tests where relatively small sensor-noise levels are selected, the two filters' performance is very similar (these results cannot be included due to limited space). One possible explanation for this is the fact that, when the sensor noise is small, the linearization errors remain relatively small for both filters, and, since both filters maintain the appropriate observability properties, their performance is comparable. However, when the noise is large, the linearization errors become significant. In this case, the statistical linearization employed by the UKF is capable of better capturing the higher-order effects in the nonlinear propagation and update models, thus yielding superior performance. Hence, we see that the OC-UKF combines the benefits of the FEJ-EKF (i.e., observability properties) with those of the UKF (i.e., better linearization), to form an estimator whose performance is comparable to that of the ideal EKF.

As a final remark, we note that even though the OC-UKF NEES performance is significantly better compared to that of the FEJ-EKF, the difference in the RMS errors of the two filters is less pronounced. This indicates that the effects of inconsistency primarily affect the covariance, rather than the state estimates.
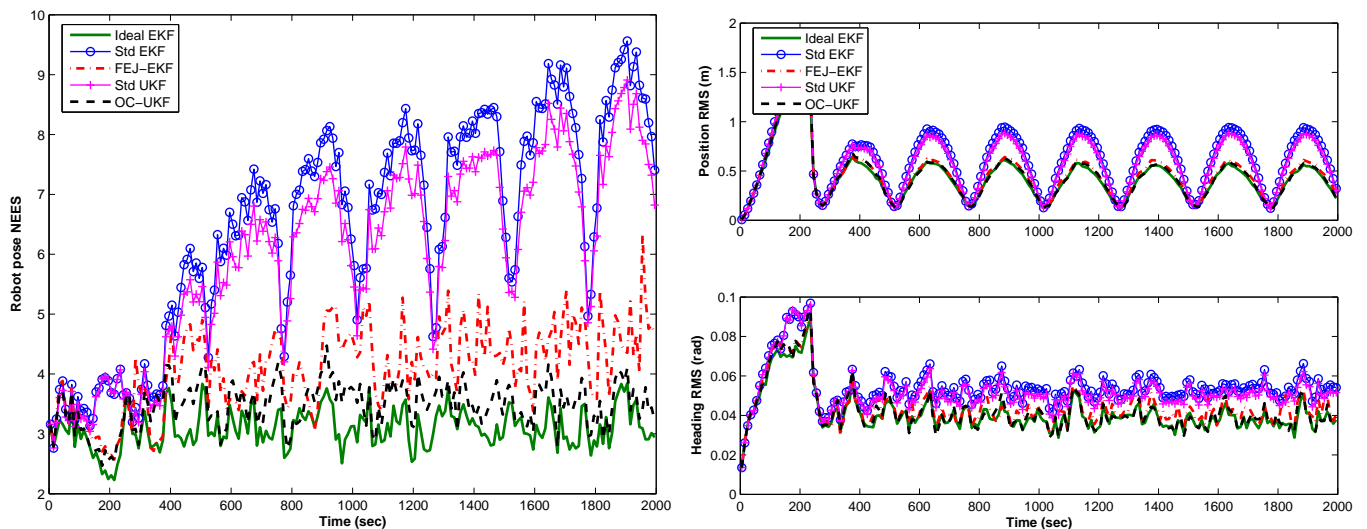
Fig. 1. Monte Carlo results for a SLAM scenario with multiple loop closures. (a) Average NEES of the robot pose errors (b) RMS errors for the robot pose (position and orientation). In these plots, the solid lines correspond to the ideal EKF, the solid lines with circles to the standard EKF, the dash-dotted lines to the FEJ-EKF, the solid lines with crosses to the standard UKF, and the dashed lines to the OC-UKF. Note that the RMS errors of the ideal EKF, the FEJ-EKF, and the OC-UKF are almost identical, which makes the corresponding lines difficult to distinguish.

VIII. CONCLUSIONS

This paper focuses on the UKF-based SLAM, and in particular on the issues of computational complexity and filter inconsistency. The first contribution of this work is a formulation of UKF-based SLAM that has computational complexity of the same order as that of EKF-SLAM. This formulation requires computing the square root of matrices of (small) *constant* size, which leads to computational complexity *linear* in the propagation phase, and *quadratic* during updates. Moreover, we have shown that a mismatch between the observability properties of the linear-regression-based system model employed in the UKF, and those of the actual nonlinear SLAM system, causes inconsistency. To address this problem, we have introduced a new Observability-Constrained (OC)-UKF, which ensures that the filter's system model has an unobservable subspace of appropriate dimensions. This filter is shown to outperform both the EKF and the UKF, in terms of consistency and accuracy.

ACKNOWLEDGEMENTS

REFERENCES

[1] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, Mar. 2000.

[2] R. Martinez-Cantin and J. A. Castellanos, "Unscented SLAM for large-scale outdoor environments," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Alberta, Canada, Aug. 2005, pp. 3427–3432.

[3] S. Holmes, G. Klein, and D. W. Murray, "A square root unscented Kalman filter for visual monoSLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, Pasadena, CA, May 2008, pp. 3710–3716.

[4] J. Langelaan and S. Rock, "Passive GPS-free navigation for small UAVs," in *Proc. of the IEEE Aerospace Conference*, Big Sky, MT, Mar. 2005, pp. 1–9.

[5] D. Chekhlov, M. Pupilli, W. Mayol-Cuevas, and A. Calway, "Real-time and robust monocular SLAM using predictive multi-resolution descriptors," in *Proc. of the 2nd International Symposium on Visual Computing*, Lake Tahoe, NV, Nov. 2006, pp. 276–285.

[6] J. Andrade-Cetto, T. Vidal-Calleja, and A. Sanfeliu, "Unscented transformation of vehicle states in SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, Apr. 2005, pp. 323–328.

[7] S. Julier and J. Uhlmann, "A counter example to the theory of simultaneous localization and map building," in *Proc. of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001, pp. 4238–4243.

[8] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 2006, pp. 3562–3568.

[9] S. Huang and G. Dissanayake, "Convergence analysis for extended Kalman filter based SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, Orlando, FL, May 2006, pp. 412–417.

[10] ——, "Convergence and consistency analysis for extended Kalman filter based SLAM," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1036–1049, Oct. 2007.

[11] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter-based SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, Pasadena, CA, May 2008, pp. 473–479.

[12] ——, "A first-estimates Jacobian EKF for improving SLAM consistency," in *Proc. of the 11th International Symposium on Experimental Robotics*, Athens, Greece, July 2008.

[13] T. Lefebvre, H. Bruyninckx, and J. De Schuller, "Comment on "a new method for the nonlinear transformation of means and covariances in filters and estimators" [and authors' reply]," *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1406–1409, Aug. 2002.

[14] G. P. Huang and S. I. Roumeliotis, "An observability constrained UKF for improving SLAM consistency," University of Minnesota, Minneapolis, MN, Tech. Rep., Aug. 2008. [Online]. Available: www.cs.umn.edu/~ghuang/paper/TR_OC-UKF.pdf

[15] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. New York: Wiley, 2001.

[16] J. Castellanos, J. Neira, and J. Tardos, "Limits to the consistency of EKF-based SLAM," in *Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, July 2004, pp. 1244–1249.