A Quadratic-Complexity Observability-Constrained Unscented Kalman Filter for 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 computational complexity in the number of states, and the inconsistency of the state estimates. To address the first issue, we introduce a new sampling strategy for the UKF, which has constant computational complexity. As a result, the overall computational complexity of UKF-based SLAM becomes of the same order as that of the extended Kalman filter (EKF)-based SLAM, i.e., quadratic in the size of the state vector. Furthermore, we investigate the inconsistency issue by analyzing the observability properties of the linear-regression-based model employed by the UKF. Based on this analysis, we propose a new algorithm, termed observability-constrained (OC)-UKF, which ensures the unobservable subspace of the UKF's linear-regression-based system model is of the same dimension as that of the nonlinear SLAM system. This results in substantial improvement in the accuracy and consistency of the state estimates. The superior performance of the OC-UKF over other state-of-the-art SLAM algorithms is validated by both Monte-Carlo simulations and real-world experiments.**

***Index Terms*—SLAM, unscented Kalman filter, computational complexity, estimator consistency, system observability**

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) remains a popular choice and has been used in many applications [1]–[3], primarily due to its relative low computational complexity and ease of implementation. However, EKF-based SLAM is vulnerable to linearization errors, which can cause poor performance or even divergence, and its state estimates are typically *inconsistent* [4]–[10]. As defined in [11], a state estimator is *consistent* if the estimation errors are zero-mean, and the estimated covariance is equal to the true covariance. Consistency is one of the primary criteria for evaluating the performance of any estimator; if an estimator is inconsistent, then the accuracy of the computed state estimates is unknown, which in turn makes the estimator unreliable. In order to reduce the estimation errors due to the EKF's linearization, the unscented Kalman filter (UKF) [12] was introduced. 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 standard (i.e., original) UKF algorithm [12] 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 (see [13]–[15]). Most importantly, empirical evidence suggests [13]–[16] that the UKF also results in inconsistent estimates in SLAM, even though its performance is better than the EKF in this respect.

Our objective in this paper is to address the aforementioned limitations of UKF-based SLAM. In particular, the main contributions of this work are the following:

- We introduce 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 scheme is provably optimal, in the sense that it minimizes the expected 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 of the same order as that of EKF-based SLAM. We stress that this new UKF sampling strategy is applicable to a large class of nonlinear estimation problems (not only the SLAM problem) where the measurements at each time step are of dimension lower than the state.
- We analytically examine the consistency of UKF-based SLAM, by studying the observability properties of the statistically-linearized (i.e., linear-regression-based) system model 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 (OC)-UKF SLAM. 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 underlying nonlinear SLAM system. As a result, the OC-UKF outperforms the standard UKF as well as

This work was supported by the Digital Technology Center (DTC) at the University of Minnesota, the UC Riverside Bourns College of Engineering, and the National Science Foundation (IIS-0643680, IIS-1117957). This paper was presented in part at the IEEE International Conference on Robotics and Automation, Kobe, Japan, May 2009.

G. P. Huang is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: ghuang@csail.mit.edu).

A. I. Mourikis is with the Department of Electrical Engineering, University of California, Riverside, CA 92521 USA (e-mail: mourikis@ee.ucr.edu).

S. I. Roumeliotis is with the Department of Computer Science and Engineering and Digital Technology Center, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: stergios@cs.umn.edu).

other state-of-the-art algorithms, in terms of accuracy and consistency, as validated by both simulation and experimental tests.

II. Related Work

The SLAM problem has received considerable attention over the past two decades. Since [17] first introduced a stochastic-mapping solution to this problem, rapid and exciting progress has been made, resulting in many competing solutions, including both filtering and smoothing approaches. In particular, filtering methods such as the EKF and the UKF recursively estimate a state vector consisting of the current robot pose and the observed landmarks [18]–[21]. Due to the fact that any (implicit or explicit) linearization-based filter marginalizes out the previous robot poses, it cannot relinearize the nonlinear system and measurement models at the past states, which may result in large linearization errors and thus degrade the filter's performance.

To better deal with nonlinearity, batch iterative optimization methods can be applied to the SLAM problem [22]–[28]. These methods, following the paradigm of bundle-adjustment (BA) algorithms originally developed in photogrammetry and computer vision [29]–[35], iteratively minimize a cost function involving the residuals of all the measurements, with respect to the entire robot trajectory and all landmarks (i.e., with no marginalization). These BA-based approaches exploit the sparsity of the measurement graph so as to speed up computation. However, for large-scale SLAM problems, a batch solution may be too computationally expensive to obtain in real time [36].

In order to reduce the computational complexity of BA, different approximate methods have been developed that either use a subset of the data to optimize over only few variables, or solve the BA problem only intermittently. Specifically, sliding-window filters (e.g., [37], [38]), compute a solution for a constant-size, sliding window of states (robot poses and landmark positions) using only the measurements corresponding to that time interval. Similarly, keyframe-based approaches (e.g., [39]–[41]), perform batch optimization over only a subset of views/keyframes. On the other hand, incremental approaches to BA such as the iSAM algorithm [42] reduce computation by employing factorization-updating methods which allow reusing the information-matrix factorization available from previous time steps. Computationally demanding procedures, such as relinearization and batch factorization, are only performed intermittently. Alternatively, the iSAM2 algorithm [36] uses the Bayes tree data structure [43], which allows for fluid or just-in-time relinearization (i.e., relinearizing only when the linearization point significantly deviates from the current estimate), as well as partial variable reordering at every update (instead of only periodic batch reordering as in iSAM [42]). Nevertheless, incremental methods can also suffer from increased computational cost, due to the accumulation of fill-in that occurs with frequent landmark re-observations.

Even though both filtering and smoothing approaches have been widely used, to this date, very little is known about which conditions favor the use of one over the other. In particular, Strasdat et al. [28], [44] recently argued that BA is, in general, better than filtering in terms of accuracy and efficiency. However, their analysis focused exclusively on the restrictive scenario of "small-scale" visual SLAM where overlapping views of the same scene are assumed over a short trajectory (less than 16 camera poses in total) and without any loop closure. Clearly, based on this limiting case study, one cannot make inferences about the relative accuracy and efficiency of filtering and smoothing algorithms in more realistic SLAM scenarios (i.e., lengthy paths with varying number of visible landmarks and loop closure events).

Although such a general study is beyond the scope of this work, in this paper we have compared the proposed OC-UKF and the state-of-the-art iSAM algorithm [42] in various SLAM scenarios, both in simulations and in real-world datasets. In particular, as shown in Sections VII and VIII, iSAM does *not* necessarily perform better than the proposed OC-UKF (in terms of estimation accuracy/consistency and computational cost). Specifically, while BA methods are certainly preferable in problems involving thousands of landmarks and few loop closures, filtering-based methods are still competitive in the case of sparser environments (e.g., tens to a few hundreds of landmarks), long-term operation, and frequent loop closures. This is due to the fact that, in the latter scenario, the computational cost of smoothing methods will continuously increase with the length of the robot's path, while the runtime of filtering algorithms will remain bounded. It is worth noting that many applications of interest (e.g., a service robot operating inside a home for an extended time period) fall under the second category.

A. UKF Computational Complexity

A number of researchers have applied the standard UKF to the SLAM problem (e.g., [13], [45], [46]). However, this requires computing the square root of the state covariance matrix at each time step, which has computational complexity *cubic* in the number of landmarks, and thus is not suitable for real-time operation in larger environments. To address this problem, Holmes et al. [14], [15] proposed the square-root UKF (SRUKF) for monocular visual SLAM, which 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 considerably more complicated implementation. Additionally, as shown in [14], [15], 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. [47] 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 pose estimate 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 proposed 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 update.

B. UKF Consistency

The *consistency* of UKF-based SLAM has received limited attention in the literature. In [13]–[15], the consistency of the UKF was empirically examined, but, to the best of our knowledge, no theoretical analysis exists to date. On the other hand, the consistency of EKF-based SLAM has been studied in a number of publications [4]–[10]. In particular, in our recent work [8]–[10], we have presented an analytical study of this issue by focusing on the observability properties of the EKF linearized system model.

In this paper, we extend this analysis to the case of UKF-based SLAM. 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. Moreover, we introduce the OC-UKF, which attains better performance than the standard UKF, by ensuring that the observability requirements on the filter's system model are satisfied. It is important to point out that, as compared to our previously-developed OC-EKF [10], the OC-UKF proposed in this paper introduces a new paradigm for computing filter Jacobians. Specifically, the OC-EKF employs a derivative-based approach to find the filter Jacobians, and subsequently optimizes the selection of linearization points. In contrast, the OC-UKF uses statistical linearization and *directly* calculates the optimal (inferred) Jacobians by solving an observability-constrained optimization (linear-regression) problem. Note also that, in comparison to our previous conference publication [16], in this paper we study in depth the observability properties of the linear-regression-based UKF system model, present in detail the derivations of the OC-UKF SLAM, and thoroughly validate its superior performance, as compared to other state-of-the-art algorithms, both in Monte-Carlo simulations and in real-world experiments.

III. LRKF AND UKF

In this section, we present the UKF in the context of the linear-regression Kalman filter (LRKF). As shown in [48], the UKF is closely related to the LRKF (with its sample points chosen deterministically, instead of randomly in the LRKF) and it can be viewed as performing an implicit *statistical linearization* of the nonlinear propagation and update models. In what follows, we 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 seeks to approximate a nonlinear function $\mathbf{y} = \mathbf{g}(\mathbf{x})$ with a linear model $\mathbf{y} \simeq \mathbf{A}\mathbf{x} + \mathbf{b}$, where $\mathbf{A}$ and $\mathbf{b}$ are the regression matrix and vector, respectively, and $\mathbf{e} \triangleq \mathbf{y} - (\mathbf{A}\mathbf{x} + \mathbf{b})$ denotes the linearization error. Once this linear approximation is computed, the LRKF proceeds by applying the regular Kalman filter equations. In particular, in computing the linear approximation of $\mathbf{g}(\mathbf{x})$, it aims to minimize the expected value of the linearization error square:

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

where $p(\mathbf{x})$ is the probability density function (pdf) of the state $\mathbf{x}$. Due to the nonlinearity of $\mathbf{y} = \mathbf{g}(\mathbf{x})$, it is generally impossible to compute the optimal solution of this minimization problem in closed form. To solve this problem, the LRKF instead first selects $r+1$ weighted sample points, $\{\mathcal{X}_i, w_i\}_{i=0}^{r}$, so that their sample mean and covariance are equal to the mean and covariance of $\mathbf{x}$:[1]

$$\bar{\mathbf{x}} = \sum_{i=0}^{r} w_i \mathcal{X}_i \ = \mathbb{E}(\mathbf{x}) \quad (2)$$

$$\bar{\mathbf{P}}_{\mathbf{xx}} = \sum_{i=0}^{r} w_i \left(\mathcal{X}_i - \bar{\mathbf{x}}\right) \left(\mathcal{X}_i - \bar{\mathbf{x}}\right)^T = \mathbb{E}\left[(\mathbf{x}-\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}})^T\right] \quad (3)$$

where $\mathbb{E}(\cdot)$ denotes the expectation operator. Then, using the sample-based approximation $p(\mathbf{x}) \simeq \sum_{i=0}^{r} w_i \delta(\mathbf{x} - \mathcal{X}_i)$, where $\delta(\cdot)$ is the Dirac delta function, the linear-regression problem (1) becomes:

$$\min_{\mathbf{A},\mathbf{b}} \ \sum_{i=0}^{r} w_i \left[\mathcal{Y}_i - (\mathbf{A}\mathcal{X}_i + \mathbf{b})\right]^T \left[\mathcal{Y}_i - (\mathbf{A}\mathcal{X}_i + \mathbf{b})\right] \quad (4)$$

where $\mathcal{Y}_i \triangleq \mathbf{g}(\mathcal{X}_i)$ are the regression points. We denote the linearization error corresponding to the sample point $\mathcal{X}_i$ by $\mathbf{e}_i \triangleq \mathcal{Y}_i - (\mathbf{A}\mathcal{X}_i + \mathbf{b})$. Note that the above cost function is identical to the one in [48], and hence the optimal solutions for $\mathbf{A}$ and $\mathbf{b}$ are given by [48]:

$$\mathbf{A} = \bar{\mathbf{P}}_{\mathbf{yx}} \bar{\mathbf{P}}_{\mathbf{xx}}^{-1} \ , \qquad \mathbf{b} = \bar{\mathbf{y}} - \mathbf{A}\bar{\mathbf{x}} \quad (5)$$

where

$$\bar{\mathbf{y}} = \sum_{i=0}^{r} w_i \mathcal{Y}_i \quad (6)$$

$$\bar{\mathbf{P}}_{\mathbf{yx}} = \sum_{i=0}^{r} w_i \left(\mathcal{Y}_i - \bar{\mathbf{y}}\right) \left(\mathcal{X}_i - \bar{\mathbf{x}}\right)^T \quad (7)$$

$$\bar{\mathbf{P}}_{\mathbf{yy}} = \sum_{i=0}^{r} w_i \left(\mathcal{Y}_i - \bar{\mathbf{y}}\right) \left(\mathcal{Y}_i - \bar{\mathbf{y}}\right)^T \quad (8)$$

In addition, using (5), (7) and (8), the sample covariance of the linearization errors is computed by:

$$\bar{\mathbf{P}}_{\mathbf{ee}} = \sum_{i=0}^{r} w_i \mathbf{e}_i \mathbf{e}_i^T = \bar{\mathbf{P}}_{\mathbf{yy}} - \mathbf{A}\bar{\mathbf{P}}_{\mathbf{xx}}\mathbf{A}^T \quad (9)$$

[1] Throughout this paper, $\bar{\mathbf{x}}$ and $\bar{\mathbf{P}}_{\mathbf{xx}}$ denote the sample mean and covariance of sample points $\mathcal{X}_i$, drawn from the pdf of the random variable $\mathbf{x}$. $\bar{\mathbf{P}}_{\mathbf{xy}}$ denotes the sample cross-correlation between the sets of samples $\mathcal{X}_i$ and $\mathcal{Y}_i$, drawn from the pdfs of the random variables $\mathbf{x}$ and $\mathbf{y}$, respectively. $\hat{\mathbf{x}}$ is used to denote the estimate of $\mathbf{x}$, and $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ is the error in this estimate. 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. Finally, $\mathbf{0}_{m\times n}$ is the $m \times n$ matrix of zeros, and $\mathbf{I}_n$ is the $n \times n$ identity matrix.

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 matrix $\mathbf{A}$ serves as an *inferred Jacobian* matrix, analogous to the Jacobian matrices in the EKF. The details are explained next.

B. LRKF Propagation

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

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \boldsymbol{o}_k) &(10)\\ &= \breve{\mathbf{\Phi}}_k \mathbf{x}_k + \breve{\mathbf{G}}_k \boldsymbol{o}_k + \mathbf{b}_k + \mathbf{e}_k &(11)\\ &= \underbrace{\begin{bmatrix}\breve{\mathbf{\Phi}}_k & \breve{\mathbf{G}}_k\end{bmatrix}}_{\mathbf{A}} \begin{bmatrix}\mathbf{x}_k \\ \boldsymbol{o}_k\end{bmatrix} + \mathbf{b}_k + \mathbf{e}_k &(12)\end{aligned}$$

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

In the LRKF propagation step, $r+1$ sample points $\{\mathcal{X}_i(k|k)\}_{i=0}^r$ are selected based on the *augmented* vector that comprises the filter state and the control input [see (12)]. The sample mean and sample covariance of $\{\mathcal{X}_i(k|k)\}_{i=0}^r$ are thus chosen as:

$$\bar{\mathbf{x}}_{k|k} = \begin{bmatrix}\hat{\mathbf{x}}_{k|k} \\ \boldsymbol{o}_{m_k}\end{bmatrix}, \quad \bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{k|k}} = \begin{bmatrix}\mathbf{P}_{k|k} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k\end{bmatrix} \qquad (13)$$

Subsequently, the LRKF produces the regression points, $\{\mathcal{Y}_i(k+1|k) = \mathbf{f}(\mathcal{X}_i(k|k))\}_{i=0}^r$, by passing the sample points through the nonlinear process function (10). The sample mean, $\bar{\mathbf{y}}_{k+1|k}$, and sample covariance, $\bar{\mathbf{P}}_{\mathbf{y}\mathbf{y}_{k+1|k}}$, of the regression points $\mathcal{Y}_i$ are used as the mean, $\hat{\mathbf{x}}_{k+1|k}$, and covariance, $\mathbf{P}_{k+1|k}$, of the propagated state estimates, respectively, i.e.,

$$\hat{\mathbf{x}}_{k+1|k} = \bar{\mathbf{y}}_{k+1|k} \ , \qquad \mathbf{P}_{k+1|k} = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{y}_{k+1|k}} \qquad (14)$$

Moreover, the inferred Jacobian matrices, $\breve{\mathbf{\Phi}}_k$ and $\breve{\mathbf{G}}_k$, which will be needed later on, are given by [see (5) and (12)]:

$$\mathbf{A} = \begin{bmatrix}\breve{\mathbf{\Phi}}_k & \breve{\mathbf{G}}_k\end{bmatrix} = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_{k|k}} \bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{k|k}}^{-1} \qquad (15)$$

where $\bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_{k|k}}$ is computed as in (7). Substituting (5) in (11) and using (13), (14) and (15), we have:

$$\begin{aligned}&\mathbf{x}_{k+1} = \breve{\mathbf{\Phi}}_k \mathbf{x}_k + \breve{\mathbf{G}}_k \boldsymbol{o}_k + \bar{\mathbf{y}}_{k+1|k} - \mathbf{A}\bar{\mathbf{x}}_{k|k} + \mathbf{e}_k\\ &= \breve{\mathbf{\Phi}}_k \mathbf{x}_k + \breve{\mathbf{G}}_k \boldsymbol{o}_k + \hat{\mathbf{x}}_{k+1|k} - \begin{bmatrix}\breve{\mathbf{\Phi}}_k & \breve{\mathbf{G}}_k\end{bmatrix} \begin{bmatrix}\hat{\mathbf{x}}_{k|k} \\ \boldsymbol{o}_{m_k}\end{bmatrix} + \mathbf{e}_k\\ &\Rightarrow \ \tilde{\mathbf{x}}_{k+1|k} = \breve{\mathbf{\Phi}}_k \tilde{\mathbf{x}}_k + \breve{\mathbf{G}}_k \mathbf{w}_k + \mathbf{e}_k \qquad (16)\end{aligned}$$

This last equation describes the linearized (based on regression) error-state propagation model used by the LRKF.

C. LRKF Update

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

$$\begin{aligned}\mathbf{z}_{k+1} &= \mathbf{h}(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1} &(17)\\ &= \breve{\mathbf{H}}_{k+1}\mathbf{x}_{k+1} + \mathbf{b}'_{k+1} + \mathbf{e}'_{k+1} + \mathbf{v}_{k+1} &(18)\end{aligned}$$

where $\mathbf{z}_{k+1}$ is the measurement and $\mathbf{v}_{k+1}$ is the zero-mean white Gaussian measurement noise, with covariance matrix $\mathbf{R}_{k+1}$. A set of $r+1$ sample points, $\{\mathcal{X}_i(k+1|k)\}_{i=0}^r$, are selected, whose sample mean and sample covariance are equal to $\hat{\mathbf{x}}_{k+1|k}$ and $\mathbf{P}_{k+1|k}$, respectively, i.e.,

$$\bar{\mathbf{x}}_{k+1|k} = \hat{\mathbf{x}}_{k+1|k} \ , \qquad \bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{k+1|k}} = \mathbf{P}_{k+1|k} \qquad (19)$$

We pass these sample points through the nonlinear measurement function (17), to obtain the regression points, $\{\mathcal{Z}_i(k+1|k) = \mathbf{h}(\mathcal{X}_i(k+1|k))\}_{i=0}^r$. The regression matrix (i.e., inferred measurement Jacobian) $\breve{\mathbf{H}}_{k+1}$is computed by [see (5)]:

$$\breve{\mathbf{H}}_{k+1} = \bar{\mathbf{P}}_{\mathbf{z}\mathbf{x}_{k+1|k}} \bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{k+1|k}}^{-1} \qquad (20)$$

where $\bar{\mathbf{P}}_{\mathbf{z}\mathbf{x}_{k+1|k}}$ is computed as in (7). Subsequently, the state and covariance are updated using the EKF update equations:

$$\begin{aligned}\mathbf{S}_{k+1} &= \bar{\mathbf{P}}_{\mathbf{z}\mathbf{z}_{k+1|k}} + \mathbf{R}_{k+1} &(21)\\ \mathbf{K}_{k+1} &= \mathbf{P}_{k+1|k}\breve{\mathbf{H}}_{k+1}^T \mathbf{S}_{k+1}^{-1} &(22)\\ \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}(\mathbf{z}_{k+1} - \bar{\mathbf{z}}_{k+1|k}) &(23)\\ \mathbf{P}_{k+1|k+1} &= \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}_{k+1}^T &(24)\end{aligned}$$

where $\bar{\mathbf{z}}_{k+1|k}$ and $\bar{\mathbf{P}}_{\mathbf{z}\mathbf{z}_{k+1|k}}$ are computed from (6) and (8), respectively.

D. UKF Sampling

In contrast to the LRKF [48], where the sample points are drawn randomly, in the UKF, $r+1 = 2n+1$ so-called sigma points $\mathcal{X}_i$ are *deterministically* chosen along with their weights w_i, $i = 1, \ldots, n$, according to the following equations [12]:

$$\begin{aligned}\mathcal{X}_0(\ell|k) &= \bar{\mathbf{x}}_{\ell|k} \ , \ w_0 = \frac{2\kappa}{2(n+\kappa)} &(25)\\ \mathcal{X}_i(\ell|k) &= \bar{\mathbf{x}}_{\ell|k} + \left[\sqrt{(n+\kappa)\bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{\ell|k}}}\right]_i \ , \ w_i = \frac{1}{2(n+\kappa)}\\ \mathcal{X}_{i+n}(\ell|k) &= \bar{\mathbf{x}}_{\ell|k} - \left[\sqrt{(n+\kappa)\bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{\ell|k}}}\right]_i \ , \ w_{i+n} = \frac{1}{2(n+\kappa)}\end{aligned}$$

where n is the dimension of $\bar{\mathbf{x}}_{\ell|k}$ [see (13) and (19)], $\left[\sqrt{(n+\kappa)\bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{\ell|k}}}\ \right]_i$ is the i-th column of the matrix $\sqrt{(n+\kappa)\bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{\ell|k}}}$, $\ell \in \{k, k+1\}$, and κ is a design parameter in the selection of the sigma points, 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 [12].

IV. Quadratic-Complexity UKF SLAM

In this section, we show how the computational cost of the UKF, when applied to the SLAM problem, can be reduced. In particular, in this paper we focus on 2D SLAM, in which the state vector consists of the robot pose (position and orientation) and the positions of M landmarks:

$$\mathbf{x}_k = \begin{bmatrix}\mathbf{x}_{R_k}^T & \mathbf{p}_L^T\end{bmatrix}^T = \begin{bmatrix}\mathbf{x}_{R_k}^T & \mathbf{p}_{L_1}^T & \cdots & \mathbf{p}_{L_M}^T\end{bmatrix}^T \tag{26}$$

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

In the UKF algorithm presented in the preceding section, the main bottleneck is the computation of the square root of the covariance matrix [see (25)], which has complexity $\mathcal{O}(M^3)$. Clearly, in a scenario where a large number of landmarks are included in the state vector, carrying out this operation during each propagation and update would incur an unacceptable computational burden. To address this problem, we here propose a new sampling scheme for the UKF, which has computational cost $\mathcal{O}(1)$, and hence reduces the complexity of the propagation and update steps to linear and quadratic, respectively. The derivation of this sampling scheme is based on the observation that, during SLAM, only a *small subset* of the state vector appears in the nonlinear propagation and measurement models. In particular, in the propagation only the robot state changes, while in the update, every measurement involves only the robot pose and one observed landmark.[2] To take advantage of this important property, we employ the following lemma:

Lemma IV.1. *Consider a nonlinear function* $\mathbf{y} = \mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}_1)$, *where only the state entries* $\mathbf{x}_1$ *of the vector* $\mathbf{x}$ *partitioned as* $\mathbf{x} = \begin{bmatrix}\mathbf{x}_1 \\ \mathbf{x}_2\end{bmatrix}$ *appear in* $\mathbf{g}(\mathbf{x})$. *Moreover, consider the regression matrix* $\mathbf{A}$ *of the linear-regression problem* (4) *accordingly partitioned as* $\mathbf{A} = \begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}$, *i.e.,*

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b} + \mathbf{e} = \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 + \mathbf{b} + \mathbf{e} \tag{27}$$

Then, the optimal solution to (4) *is:*

$$\mathbf{A}_1 = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}, \qquad \mathbf{A}_2 = \mathbf{0}, \qquad \mathbf{b} = \bar{\mathbf{y}} - \mathbf{A}_1\hat{\mathbf{x}}_1 \tag{28}$$

Proof. See Appendix A. □

This lemma shows that, in order to minimize the expected squared error of the statistical linearization (4), it suffices to draw sample points from the pdf of $\mathbf{x}_1$. As mentioned before, in SLAM the number of states participating in the nonlinear process and measurement models is constant. 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 whole state. Compared to the EKF-SLAM, the proposed UKF-SLAM only incurs a small computational overhead (for computing the square roots of constant-size matrices), and has computational complexity of the same order. In the following, we present in detail this new sampling strategy used in UKF-SLAM. We stress again that apart from the particular problem of SLAM treated in this paper, this new UKF sampling scheme is applicable to *any* problem where the measurements only involve a subset of the state vector.

A. Propagation

During propagation, only the robot pose and the control input (odometry) participate in the process model [see (10)]. 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. The resulting Jacobians are then used for efficiently propagating the covariance matrix corresponding to the entire state.

We start by drawing the sigma points $\mathcal{X}_i(k|k)$ based on the vector with the following mean and covariance [see (13)]:

$$\bar{\mathbf{x}}_{k|k} = \begin{bmatrix}\hat{\mathbf{x}}_{R_{k|k}} \\ \boldsymbol{o}_{m_k}\end{bmatrix}, \qquad \bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{k|k}} = \begin{bmatrix}\mathbf{P}_{RR_{k|k}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k\end{bmatrix} \tag{29}$$

where $\mathbf{P}_{RR_{k|k}}$ is the covariance matrix corresponding to the robot pose, obtained by partitioning the state covariance matrix as follows [see (26)]:

$$\mathbf{P}_{k|k} = \begin{bmatrix}\mathbf{P}_{RR_{k|k}} & \mathbf{P}_{RL_{k|k}} \\ \mathbf{P}_{RL_{k|k}}^T & \mathbf{P}_{LL_{k|k}}\end{bmatrix} \tag{30}$$

Note that the vector $\bar{\mathbf{x}}_{k|k}$ in (29) is of dimension $n = 5$ (assuming that the odometry measurement $\boldsymbol{o}_{m_k}$ is two-dimensional), and thus the computational cost of computing the sigma points is very low.

Subsequently, we transform the sigma points, $\{\mathcal{X}_i(k|k)\}_{i=0}^{10}$, using the process model (10), to obtain the regression points of the propagated robot pose, $\{\mathcal{Y}_i(k+1|k) = \mathbf{f}(\mathcal{X}_i(k|k))\}_{i=0}^{10}$. This enables us to compute the mean, $\hat{\mathbf{x}}_{R_{k+1|k}} = \bar{\mathbf{y}}_{k+1|k}$, and covariance, $\mathbf{P}_{RR_{k+1|k}} = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{y}_{k+1|k}}$, of the propagated robot pose, in the same way as in the standard LRKF/UKF [see (14)]. Moreover, we can evaluate the *inferred* robot state and odometry Jacobians as [see (28) and (15)]:

$$\mathbf{A}_1 = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_{k|k}}\bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{k|k}}^{-1} = \begin{bmatrix}\breve{\mathbf{\Phi}}_{R_k} & \breve{\mathbf{G}}_{R_k}\end{bmatrix} \tag{31}$$

while $\mathbf{A}_2 = \mathbf{0}$.

Next, using (16), we compute the propagated cross-correlation between the robot and the landmarks as follows:

$$\begin{aligned}\mathbf{P}_{RL_{k+1|k}} &= \mathbb{E}\left[\tilde{\mathbf{x}}_{R_{k+1|k}}\tilde{\mathbf{p}}_{L_{k|k}}^T\right] \\ &= \mathbb{E}\left[\left(\breve{\mathbf{\Phi}}_{R_k}\tilde{\mathbf{x}}_{R_{k|k}} + \breve{\mathbf{G}}_{R_k}\mathbf{w}_k + \mathbf{e}_k\right)\tilde{\mathbf{p}}_{L_{k|k}}^T\right] \\ &= \breve{\mathbf{\Phi}}_{R_k}\mathbf{P}_{RL_{k|k}}\end{aligned} \tag{32}$$

Thus, the propagated state covariance matrix is given by:

$$\mathbf{P}_{k+1|k} = \begin{bmatrix}\bar{\mathbf{P}}_{\mathbf{y}\mathbf{y}_{k+1|k}} & \breve{\mathbf{\Phi}}_{R_k}\mathbf{P}_{RL_{k|k}} \\ \mathbf{P}_{RL_{k|k}}^T\breve{\mathbf{\Phi}}_{R_k}^T & \mathbf{P}_{LL_{k|k}}\end{bmatrix} \tag{33}$$

which is evaluated at a cost only *linear* in the size of the state vector, similarly to the EKF.

The matrix $\breve{\mathbf{\Phi}}_{R_k}$ derived in (31) is the inferred propagation Jacobian for the robot state. To compute the inferred Jacobian matrix for the entire SLAM state vector, which will be useful for our ensuing analysis, we use (9), (29) and (31) to write

$$\begin{aligned}\bar{\mathbf{P}}_{\mathbf{y}\mathbf{y}_{k+1|k}} &= \mathbf{A}_1\bar{\mathbf{P}}_{\mathbf{x}\mathbf{x}_{k|k}}\mathbf{A}_1^T + \bar{\mathbf{P}}_{\mathbf{e}\mathbf{e}_k} \\ &= \breve{\mathbf{\Phi}}_{R_k}\mathbf{P}_{RR_{k|k}}\breve{\mathbf{\Phi}}_{R_k}^T + \breve{\mathbf{G}}_{R_k}\mathbf{Q}_k\breve{\mathbf{G}}_{R_k}^T + \bar{\mathbf{P}}_{\mathbf{e}\mathbf{e}_k}\end{aligned} \tag{34}$$

and therefore, (33) can equivalently be written as:

$$\mathbf{P}_{k+1|k} = \breve{\mathbf{\Phi}}_k\mathbf{P}_{k|k}\breve{\mathbf{\Phi}}_k^T + \mathbf{Q}_k^* \tag{35}$$

[2] When more than one landmarks are detected concurrently, their measurements can be processed sequentially, given that the measurement noise in different observations is independent.

where

$$\mathbf{Q}_k^* = \begin{bmatrix} \breve{\mathbf{G}}_{R_k}\mathbf{Q}_k\breve{\mathbf{G}}_{R_k}^T + \bar{\mathbf{P}}_{\mathbf{ee}_k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{36}$$

$$\breve{\mathbf{\Phi}}_k = \begin{bmatrix} \breve{\mathbf{\Phi}}_{R_k} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{2M} \end{bmatrix} \tag{37}$$

In the above expression, $\breve{\mathbf{\Phi}}_k$ is the inferred propagation Jacobian matrix for the SLAM state vector comprising the robot pose and the landmark positions.

B. Update

Measurements used for updates involve only the robot pose and the position of one observed landmark. Therefore, we can apply the unscented transformation only to this subset of states so as to reduce the computational cost. In particular, assume that the j-th landmark, L_j, is observed at time-step $k+1$. Then, the set of sigma points $\{\mathcal{X}_i(k+1|k)\}_{i=0}^{10}$ are drawn from a distribution with the following mean and covariance:

$$\bar{\mathbf{x}}_{k+1|k} = \begin{bmatrix} \hat{\mathbf{x}}_{R_{k+1|k}} \\ \hat{\mathbf{p}}_{L_{j,k+1|k}} \end{bmatrix}, \quad \bar{\mathbf{P}}_{\mathbf{xx}_{k+1|k}} = \begin{bmatrix} \mathbf{P}_{RR_{k+1|k}} & \mathbf{P}_{RL_{j,k+1|k}} \\ \mathbf{P}_{L_jR_{k+1|k}} & \mathbf{P}_{L_jL_{j,k+1|k}} \end{bmatrix} \tag{38}$$

where $\mathbf{P}_{RR_{k+1|k}}$ and $\mathbf{P}_{L_jL_{j,k+1|k}}$ are the covariance matrices of the robot and the landmark, respectively, while $\mathbf{P}_{RL_{j,k+1|k}} = \mathbf{P}_{L_jR_{k+1|k}}^T$ is the corresponding cross-correlation matrix, obtained from the following partitioning of the state covariance matrix:

$$\mathbf{P}_{k+1|k} = \begin{bmatrix} \mathbf{P}_{RR_{k+1|k}} & \cdots & \mathbf{P}_{RL_{j,k+1|k}} & \cdots & \mathbf{P}_{RL_{M,k+1|k}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{P}_{L_jR_{k+1|k}} & \cdots & \mathbf{P}_{L_jL_{j,k+1|k}} & \cdots & \mathbf{P}_{L_jL_{M,k+1|k}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{P}_{L_MR_{k+1|k}} & \cdots & \mathbf{P}_{L_ML_{j,k+1|k}} & \cdots & \mathbf{P}_{L_ML_{M,k+1|k}} \end{bmatrix}$$

Note that the matrix used for generating the sigma points has constant size [see (38)], regardless of the number of landmarks in the state vector.

Once the set of sigma points are generated, the linear regression of the LRKF update (see Section III-C) is applied to obtain the *inferred* measurement Jacobian for the pertinent states [see (20) and (28)]:

$$\mathbf{A}_1 = \bar{\mathbf{P}}_{\mathbf{zx}_{k+1|k}}\bar{\mathbf{P}}_{\mathbf{xx}_{k+1|k}}^{-1} = \begin{bmatrix} \breve{\mathbf{H}}_{R_{k+1}} & \breve{\mathbf{H}}_{L_{j,k+1}} \end{bmatrix} \tag{39}$$

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 j-th landmark. To construct the inferred measurement Jacobian for the entire state vector, we note that according to the new sampling scheme the unscented transformation is not applied to the landmarks that are not currently observed (their regression matrices are zero according to Lemma IV.1). Therefore, the *inferred* measurement Jacobian for the entire SLAM state vector is:

$$\breve{\mathbf{H}}_{k+1} = \begin{bmatrix} \breve{\mathbf{H}}_{R_{k+1}} & \mathbf{0} & \cdots & \mathbf{0} & \breve{\mathbf{H}}_{L_{j,k+1}} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \tag{40}$$

Once this matrix is available, (21)-(24) are applied to update the state estimate and covariance in the UKF. It is important to point out that the computational cost of the proposed UKF update equations is dominated by the covariance update (24), and hence is quadratic in the number of landmarks, similarly to the EKF.

C. Landmark Initialization

Suppose that the j-th landmark, L_j, is first observed at time-step k_o. The corresponding measurement is given by:[3]

$$\mathbf{z}_{k_o} = \mathbf{h}(\mathbf{x}_{R_{k_o}}, \mathbf{p}_{L_j}) + \mathbf{v}_{k_o} = \mathbf{z}_{k_o}^* + \mathbf{v}_{k_o} \tag{41}$$

By solving $\mathbf{z}_{k_o}^* = \mathbf{h}(\mathbf{x}_{R_{k_o}}, \mathbf{p}_{L_j})$ for $\mathbf{p}_{L_j}$, we can express the landmark position as a (generally nonlinear) function of the robot pose and the noiseless measurement:

$$\mathbf{p}_{L_j} = \mathbf{g}(\mathbf{x}_{R_{k_o}}, \mathbf{z}_{k_o}^*) \tag{42}$$

In order to carry out the landmark initialization in the LRKF/UKF framework, we approximate this nonlinear initialization function by a linear function:

$$\begin{aligned} \mathbf{p}_{L_j} &= \mathbf{A}_\mathbf{x}\mathbf{x}_{R_{k_o}} + \mathbf{A}_\mathbf{z}\mathbf{z}_{k_o}^* + \mathbf{b}_{k_o}'' + \mathbf{e}_{k_o}'' & (43) \\ &= \underbrace{\begin{bmatrix} \mathbf{A}_\mathbf{x} & \mathbf{A}_\mathbf{z} \end{bmatrix}}_{\mathbf{A}_1} \begin{bmatrix} \mathbf{x}_{R_{k_o}} \\ \mathbf{z}_{k_o}^* \end{bmatrix} + \mathbf{b}_{k_o}'' + \mathbf{e}_{k_o}'' & (44) \end{aligned}$$

where $\mathbf{A}_\mathbf{x}$ and $\mathbf{A}_\mathbf{z}$ are the regression matrices corresponding to the robot pose and the measurement, respectively. These matrices are computed by statistical linearization, similarly to the cases of propagation and update.

Specifically, it becomes clear from (42) that only the robot pose and the measurement of the newly detected landmark are involved in the initialization process. Therefore, we can apply the result of Lemma IV.1 to draw the sigma points based on the vector with the following mean and covariance:

$$\bar{\mathbf{x}}_{k_o|k_o-1} = \begin{bmatrix} \hat{\mathbf{x}}_{R_{k_o|k_o-1}} \\ \mathbf{z}_{k_o} \end{bmatrix}, \quad \bar{\mathbf{P}}_{\mathbf{xx}_{k_o|k_o-1}} = \begin{bmatrix} \mathbf{P}_{RR_{k_o|k_o-1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{k_o} \end{bmatrix} \tag{45}$$

Suppose the measurement vector is of dimension m. Then, the UKF chooses $r+1 = 2 \times (3+m) + 1$ sigma points, $\{\mathcal{X}_i(k_o|k_o-1)\}_{i=0}^{r}$, and transforms them through the nonlinear initialization model (42) to obtain the regression points of the new landmark position $\{\mathcal{Y}_i(k_o|k_o-1) = \mathbf{g}(\mathcal{X}_i(k_o|k_o-1))\}_{i=0}^{r}$. The sample mean of the regression points is used to initialize the new landmark position:

$$\hat{\mathbf{p}}_{L_{j,k_o|k_o}} = \bar{\mathbf{y}}_{k_o|k_o-1} \tag{46}$$

In order to compute the covariance matrix of the augmented state vector comprising the robot pose, the previously initialized landmarks, and the new landmark, we first note that the regression matrix in (44) is [see (28)]:

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{A}_\mathbf{x} & \mathbf{A}_\mathbf{z} \end{bmatrix} = \bar{\mathbf{P}}_{\mathbf{yx}_{k_o|k_o-1}}\bar{\mathbf{P}}_{\mathbf{xx}_{k_o|k_o-1}}^{-1} \tag{47}$$

Subsequently, using (43), (45), (46), and (5), we compute the error in the posterior estimate for the position of the j-th landmark:

$$\begin{aligned} &\tilde{\mathbf{p}}_{L_{j,k_o|k_o}} = \mathbf{p}_{L_j} - \hat{\mathbf{p}}_{L_{j,k_o|k_o}} \\ &= \mathbf{A}_\mathbf{x}\mathbf{x}_{R_{k_o}} + \mathbf{A}_\mathbf{z}\mathbf{z}_{k_o}^* + \mathbf{b}_{k_o}'' + \mathbf{e}_{k_o}'' - \mathbf{A}_\mathbf{x}\hat{\mathbf{x}}_{R_{k_o|k_o-1}} - \mathbf{A}_\mathbf{z}\mathbf{z}_{k_o} - \mathbf{b}_{k_o}'' \\ &= \mathbf{A}_\mathbf{x}\tilde{\mathbf{x}}_{R_{k_o|k_o-1}} + \mathbf{A}_\mathbf{z}\mathbf{v}_{k_o} + \mathbf{e}_{k_o}'' \end{aligned} \tag{48}$$

[3] To preserve the clarity of presentation, we consider the case where a single measurement suffices to initialize the landmark. This includes the distance-bearing measurement model, commonly used in practice. However, this is not a necessary assumption and our analysis can be extended to the case where multiple measurements at different time steps are needed to initialize the new landmark (e.g., bearing-only or distance-only measurements).

Based on (48), the cross-correlation terms between the new landmark and the robot and the old landmarks are given by:

$$\mathbf{P}_{L_j R_{k_o|k_o}} = \mathbb{E}\left[\tilde{\mathbf{p}}_{L_{j,k_o|k_o}} \tilde{\mathbf{x}}^T_{R_{k_o|k_o-1}}\right] = \mathbf{A}_{\mathbf{x}} \mathbf{P}_{RR_{k_o|k_o-1}} \tag{49}$$

$$\mathbf{P}_{L_j L_{j',k_o|k_o}} = \mathbb{E}\left[\tilde{\mathbf{p}}_{L_{j,k_o|k_o}} \tilde{\mathbf{p}}^T_{L_{j',k_o|k_o-1}}\right] = \mathbf{A}_{\mathbf{x}} \mathbf{P}_{RL_{j',k_o|k_o-1}} \tag{50}$$

for $j' = 1, \dots, M$ and $j' \neq j$. Hence, the covariance matrix of the augmented state vector becomes:

$$\mathbf{P}_{k_o|k_o} = \begin{bmatrix} \mathbf{P}_{k_o|k_o-1} & \mathbf{P}_{k_o|k_o-1} \begin{bmatrix} \mathbf{A}_{\mathbf{x}}^T \\ \mathbf{0}_{2M\times 2} \end{bmatrix} \\ \begin{bmatrix} \mathbf{A}_{\mathbf{x}} & \mathbf{0}_{2\times 2M} \end{bmatrix} \mathbf{P}_{k_o|k_o-1} & \mathbf{P}_{L_j L_{j,k_o|k_o}} \end{bmatrix} \tag{51}$$

where $\mathbf{P}_{L_j L_{j,k_o|k_o}} = \bar{\mathbf{P}}_{\mathbf{yy}_{k_o|k_o-1}}$ is the sample covariance of the set of the regression points $\{\mathcal{Y}_i(k_o|k_o - 1)\}_{i=0}^r$. Note that the computational complexity of the UKF landmark initialization is linear in the number of landmarks, which is of the same order as in the EKF.

For our derivations in the following sections, it will be necessary to compute the inferred measurement Jacobian matrices, $\breve{\mathbf{H}}_{R_{k_o}}$ and $\breve{\mathbf{H}}_{L_{j,k_o}}$, which correspond to the measurement used for initializing the landmark. For this purpose, by solving (43) for $\mathbf{z}^*_{k_o}$ and then substituting in (41), we have:

$$\mathbf{z}_{k_o} = -\mathbf{A}_{\mathbf{z}}^{-1}\mathbf{A}_{\mathbf{x}}\mathbf{x}_{R_{k_o}} + \mathbf{A}_{\mathbf{z}}^{-1}\mathbf{p}_{L_j} - \mathbf{A}_{\mathbf{z}}^{-1}\mathbf{b}''_{k_o} - \mathbf{A}_{\mathbf{z}}^{-1}\mathbf{e}''_{k_o} + \mathbf{v}_{k_o} \tag{52}$$

We thus conclude that the *inferred* measurement Jacobians corresponding to this measurement are:

$$\breve{\mathbf{H}}_{R_{k_o}} = -\mathbf{A}_{\mathbf{z}}^{-1}\mathbf{A}_{\mathbf{x}} \ , \quad \breve{\mathbf{H}}_{L_{j,k_o}} = \mathbf{A}_{\mathbf{z}}^{-1} \tag{53}$$

V. SLAM Observability Analysis

As discussed in Section III, the UKF carries out recursive state estimation based on a linear approximation (i.e., using sigma points) of the nonlinear system model. In this section, we examine the *observability* properties of the UKF linear-regression-based system model, since they can affect the filter's performance. To the best of our knowledge, no such analysis has appeared in the literature prior to [16].

A. Background

Our motivation arises from our previous work [8]–[10], where it was shown that the observability properties of the EKF's linearized system model greatly impact the filter's consistency in SLAM. Specifically, we have proven in [8]–[10] that the system model of an *ideal* EKF, whose Jacobians are evaluated at the *true* state, has *3 unobservable degrees of freedom (d.o.f.)*. These correspond to the global position and orientation, and match the unobservable directions of the underlying nonlinear SLAM system [10], [49]. Moreover, it was shown that the ideal EKF exhibits excellent performance in terms of consistency. By contrast, the system model of the (standard) EKF, which uses the *current* state estimates for computing the Jacobians, has *only 2 unobservable d.o.f.*, corresponding to the global position. As a result, the standard EKF becomes inconsistent since it acquires non-existent information along the direction of the global orientation. Based on this analysis, in [8], [9], we derived the First-Estimates Jacobian (FEJ)-EKF, which, by evaluating the Jacobians at the first available state estimates, achieves the desired observability properties (i.e., its system model has *3 unobservable d.o.f.*). However, the first state estimates may be inaccurate and result in large linearization errors, thus degrading the filter's performance. To improve the FEJ-EKF, in [10], we developed the Observability-Constrained (OC)-EKF which instead selects linearization points that not only ensure the linearized system model has the *correct number of unobservable d.o.f.*, but also *minimize the linearization errors*. As a result, the OC-EKF attains consistency better than that of the FEJ-EKF and comparable to that of the ideal EKF.

In this work, we adopt an analogous approach where we first examine the observability properties of the UKF-SLAM system model and compare them to those of the underlying nonlinear SLAM system. Based on this analysis, we introduce an efficient algorithm for computing the appropriate *inferred* measurement Jacobians that preserve the dimensions of the unobservable subspace, thus improving consistency.

B. UKF-SLAM Observability

To examine the observability properties of the UKF-SLAM system model, we form the observability matrix [50] for the time interval $[k_o, k_o + k]$ as follows:

$$\mathbf{M} = \begin{bmatrix} \breve{\mathbf{H}}_{k_o} \\ \breve{\mathbf{H}}_{k_o+1}\breve{\mathbf{\Phi}}_{k_o} \\ \vdots \\ \breve{\mathbf{H}}_{k_o+k}\breve{\mathbf{\Phi}}_{k_o+k-1}\cdots\breve{\mathbf{\Phi}}_{k_o} \end{bmatrix} \tag{54}$$

where the inferred measurement Jacobians, $\breve{\mathbf{H}}_{k_o+\ell}$, $\ell \in \{0, \dots, k\}$, and inferred state propagation Jacobians, $\breve{\mathbf{\Phi}}_{k_o+\ell-1}$, $\ell \in \{1, \dots, k\}$, are computed based on the UKF regression matrices [see (31), (37), (39), and (40)].

Since the UKF approximates the nonlinear SLAM model by a regression-based linearized system [see (12) and (18)], it is desirable that its observability properties match those of the underlying nonlinear system. That is, the UKF-SLAM system model should have 3 unobservable d.o.f., or equivalently its observability matrix, $\mathbf{M}$, should have a nullspace of dimension 3.

However, this is generally *not* the case. In fact, when numerically computing the dimension of the nullspace of $\mathbf{M}$, we find that it is 3 only at time-step k_o, when a landmark is initialized. At that time, the observability matrix comprises only the first inferred measurement Jacobian, i.e., $\mathbf{M} = \breve{\mathbf{H}}_{k_o}$, which is a 2 × 5 matrix and thus generally has a nullspace of dimension 3. Later on and as more measurements become available, the dimension of the nullspace of the observability matrix decreases fast. Typically, the observability matrix $\mathbf{M}$ becomes full-rank after two time steps of consecutive observations.

A full-rank observability matrix indicates that the linear-regression-based system model employed by the UKF is observable, which contradicts the observability analysis of the nonlinear SLAM system [10], [49]. In practice, this implies that the UKF obtains "spurious" information, in all directions of the state space, even in directions where no information is available, such as the global position and orientation. This, in turn, leads to an unjustified reduction of the state estimates' covariance matrix, which cannot be compensated for by the noise covariance increase that the UKF uses to account for linearization errors [see (35)]. As shown in the simulation and experimental results in Sections VII and VIII, the inconsistency due to the mismatch between the observability properties of the UKF linear-regression-based system model and the nonlinear (or equivalently the ideal EKF) system model, causes a significant degradation in the filter's performance.

VI. Observability-Constrained UKF SLAM

In this section, we introduce a novel OC-UKF algorithm that employs a linear-regression-based system model with observability properties similar to those of the underlying nonlinear SLAM system. Specifically, we construct the "inferred" Jacobians of the UKF in such a way that the resulting system model has an *unobservable subspace of dimension 3*.

In particular, 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 (4) for computing the regression matrix, we formulate a *constrained* minimization problem that enforces the desired observability properties. Specifically, if the first landmark was observed at time-step k_o, we require that [see (54)]:

$$\mathbf{M}\mathbf{N} = \mathbf{0} \;\Leftrightarrow\; \begin{cases} \breve{\mathbf{H}}_{k_o}\mathbf{N} = \mathbf{0}, \text{ for } \ell = 0 \\ \breve{\mathbf{H}}_{k_o+\ell}\breve{\mathbf{\Phi}}_{k_o+\ell-1}\cdots\breve{\mathbf{\Phi}}_{k_o}\mathbf{N} = \mathbf{0}, \text{ for } \ell > 0 \end{cases} \tag{55}$$

In the above expressions, $\mathbf{N}$ is a $(3+2M)\times 3$ matrix, whose columns span the desired nullspace. These constraints ensure that all the block rows of the observability matrix $\mathbf{M}$ (54) have the *same* nullspace, which coincides with the *unobservable subspace* of the filter's system model. By ensuring that its inferred system model has an unobservable subspace of dimension 3, the OC-UKF avoids the infusion of erroneous information, and is empirically shown to attain significantly improved consistency (see Sections VII and VIII).

In what follows, we show how the nullspace matrix $\mathbf{N}$ is determined, and based on that, we compute the inferred measurement Jacobians.

A. Computing the Nullspace Matrix $\mathbf{N}$

Consider the following partitioning of the matrix $\mathbf{N}$:

$$\mathbf{N} = \begin{bmatrix}\mathbf{N}_R^T & \mathbf{N}_{L_1}^T & \cdots & \mathbf{N}_{L_M}^T\end{bmatrix}^T \tag{56}$$

where $\mathbf{N}_R$ is a 3×3 submatrix corresponding to the robot pose, and $\mathbf{N}_{L_i}$, $i = 1,\ldots,M$, are 2×3 submatrices corresponding to the i-th landmark. It is important to note that landmarks are typically observed and initialized at different time instants, and hence the number of submatrices comprising $\mathbf{N}$ will increase over time, as new landmarks are included into the state vector.

1) Initialization of the first landmark: When the first landmark is initialized at time-step k_o, we choose $\mathbf{N}$ to be a matrix whose columns span the nullspace of the 2×5 inferred Jacobian $\breve{\mathbf{H}}_{k_o} = \begin{bmatrix}\breve{\mathbf{H}}_{R_{k_o}} & \breve{\mathbf{H}}_{L_{1,k_o}}\end{bmatrix}$ [see (53)], i.e.,

$$\text{null}(\breve{\mathbf{H}}_{k_o}) = \underset{\text{col.}}{\text{span}}(\mathbf{N}) \Rightarrow \begin{bmatrix}\breve{\mathbf{H}}_{R_{k_o}} & \breve{\mathbf{H}}_{L_{1,k_o}}\end{bmatrix}\begin{bmatrix}\mathbf{N}_R \\ \mathbf{N}_{L_1}\end{bmatrix} = \mathbf{0} \tag{57}$$

Thus, $\mathbf{N}$ can be readily computed via the singular value decomposition (SVD) of $\breve{\mathbf{H}}_{k_o}$ [51].

2) Initialization of subsequent landmarks: Suppose that the j-th landmark is detected for the first time at time-step k_o+k. This implies that the state vector already contains the first $(j-1)$ landmarks and thus $\mathbf{N}_R$ and $\mathbf{N}_{L_i}$ $(i = 1,\ldots,j-1)$ have been computed. The nullspace matrix $\mathbf{N}$ now will have to be augmented by $\mathbf{N}_{L_j}$, corresponding to the new landmark, L_j. To determine $\mathbf{N}_{L_j}$, we first notice that, based on the structure of the measurement and state-propagation inferred Jacobians [see (40) and (37)], the corresponding block row of the observability matrix at this time step, denoted by $\mathbf{M}_{k_o+k}$, can be obtained as [see (54)]:

$$\begin{aligned}&\mathbf{M}_{k_o+k} \triangleq \breve{\mathbf{H}}_{k_o+k}\breve{\mathbf{\Phi}}_{k_o+k-1}\cdots\breve{\mathbf{\Phi}}_{k_o} = \\ &\begin{bmatrix}\breve{\mathbf{H}}_{R_{k_o+k}}\breve{\mathbf{\Phi}}_{R_{k_o+k-1}}\cdots\breve{\mathbf{\Phi}}_{R_{k_o}} & \mathbf{0} & \cdots & \mathbf{0} & \breve{\mathbf{H}}_{L_{j,k_o+k}}\end{bmatrix}\end{aligned} \tag{58}$$

Since this is the newest landmark, it is appended at the end of the state vector. Then, we compute $\mathbf{N}_{L_j}$ based on the requirement that each block row of the observability matrix $\mathbf{M}$ has the same nullspace, spanned by $\mathbf{N}$, i.e.,

$$\text{null}(\mathbf{M}_{k_o+k}) = \underset{\text{col.}}{\text{span}}(\mathbf{N}) \;\Rightarrow\; \mathbf{M}_{k_o+k}\mathbf{N} = \mathbf{0} \tag{59}$$

Substitution of (56) (using $M = j$) and (58) in (59) yields:

$$\begin{aligned}&\breve{\mathbf{H}}_{R_{k_o+k}}\breve{\mathbf{\Phi}}_{R_{k_o+k-1}}\cdots\breve{\mathbf{\Phi}}_{R_{k_o}}\mathbf{N}_R + \breve{\mathbf{H}}_{L_{j,k_o+k}}\mathbf{N}_{L_j} = \mathbf{0} \;\Rightarrow \\ &\mathbf{N}_{L_j} = -\breve{\mathbf{H}}^{-1}_{L_{j,k_o+k}}\breve{\mathbf{H}}_{R_{k_o+k}}\breve{\mathbf{\Phi}}_{R_{k_o+k-1}}\cdots\breve{\mathbf{\Phi}}_{R_{k_o}}\mathbf{N}_R\end{aligned} \tag{60}$$

B. Computing the Inferred Measurement Jacobians

We now show how the inferred measurement Jacobians for the observation of landmark L_j at time-step $k_o+\ell$ can be computed. From Lemma IV.1, we know that for these Jacobians we only need to determine the regression matrix $\mathbf{A}_1$, instead of the full regression matrix $\mathbf{A}$ [see (39) and (40)]. In the OC-UKF, at each update step after L_j has been initialized, we formulate the following *constrained* linear-regression problem with respect to $\mathbf{A}_1$ and $\mathbf{b}$ [see (4)]:

$$\min_{\mathbf{A}_1,\mathbf{b}} \quad \sum_{i=0}^{10} w_i\left[\mathcal{Z}_i - (\mathbf{A}_1\mathcal{X}_i + \mathbf{b})\right]^T\left[\mathcal{Z}_i - (\mathbf{A}_1\mathcal{X}_i + \mathbf{b})\right] \tag{61}$$

$$\text{s. t.} \quad \mathbf{A}_1\breve{\mathbf{\Phi}}'_{k_o+k-1}\cdots\breve{\mathbf{\Phi}}'_{k_o}\mathbf{N}_j = \mathbf{0} \tag{62}$$

where $\breve{\mathbf{\Phi}}'_{k_o+\ell} \triangleq \begin{bmatrix}\breve{\mathbf{\Phi}}_{R_{k_o+\ell}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_2\end{bmatrix}$, $\ell = 0,\ldots,k-1$, is the reduced-size regression matrix obtained from propagation [see (31) and (37)], corresponding to the part of the state comprising only the robot pose and L_j; and $\mathbf{N}_j \triangleq \begin{bmatrix}\mathbf{N}_R^T & \mathbf{N}_{L_j}^T\end{bmatrix}^T$ contains the corresponding block rows of $\mathbf{N}$ [see (56)]. The sigma points used in the minimization problem (61) are computed by the procedure described in Section IV-B. The optimal solution of $\mathbf{A}_1$ is obtained in *closed form* using the following lemma:

Lemma VI.1. *The optimal solution to the constrained minimization problem* (61)-(62) *is given by:*

$$\begin{aligned}\mathbf{A}_1 &= \begin{bmatrix}\breve{\mathbf{H}}_{R_{k_o+k}} & \breve{\mathbf{H}}_{L_{j,k_o+k}}\end{bmatrix} \\ &= \bar{\mathbf{P}}_{\mathbf{zx}_{k_o+k|k_o+k-1}}\mathbf{L}^T\left(\mathbf{L}\bar{\mathbf{P}}_{\mathbf{xx}_{k_o+k|k_o+k-1}}\mathbf{L}^T\right)^{-1}\mathbf{L}\end{aligned} \tag{63}$$

with

$$\mathbf{L} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{m\times(5-m)} \end{bmatrix} \left(\mathbf{I}_5 - \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T \right) \tag{64}$$

$$\mathbf{U} = \breve{\mathbf{\Phi}}'_{k_o+k-1} \cdots \breve{\mathbf{\Phi}}'_{k_o} \mathbf{N}_j \tag{65}$$

where m is the dimension of the measurement vector.

Proof. See Appendix B. □

Lastly, once we construct the full inferred measurement Jacobian matrix $\breve{\mathbf{H}}_{k_o+k}$ in (40) from the regression matrix $\mathbf{A}_1$ in (63), we update the state estimate and covariance based on (21)-(24). In summary, the main steps of the OC-UKF SLAM are outlined in Algorithm 1.

Algorithm 1 Observability-Constrained (OC)-UKF SLAM

```
Require: Initial state estimate and covariance
loop
    Propagation: When an odometry measurement is received,
    determine sigma points by (25) with mean and covariance (29).
    produce regression points by passing the sigma points through (10).
    compute the state estimate via (14).
    compute the regression matrix via (31).
    compute the propagated covariance via (33).

    Update: When a robot-to-landmark measurement is received,
    determine sigma points by (25) with mean and covariance (38).
    produce regression points by passing the sigma points through the nonlinear function (17).
    compute the regression matrix via (63) and (39).
    update the state and covariance via (21)-(24).

    Initialization: When a new landmark is detected,
    determine sigma points by (25) with mean and covariance (45).
    produce regression points by passing the sigma points through (42).
    compute the inferred Jacobian matrices via (53).
    initialize the new landmark position (46) and update the state covariance (51).
    if this is the first observed landmark, compute the nullspace matrix $\mathbf{N}$ via (57), else augment the nullspace matrix $\mathbf{N}$ with $\mathbf{N}_{L_j}$, corresponding to the new landmark (60).
end loop
```

We stress that if multiple landmarks are observed concurrently, the above process for determining the inferred measurement Jacobians is repeated sequentially for each of the landmarks. Note also that the maximum dimension of all the matrices involved in (63)-(64) is 5 [see (38)], and thus computing the regression matrix $\mathbf{A}_1$ incurs only a *constant* computational overhead, regardless of the number of landmarks in the state. As a result, the overall computational cost of the OC-UKF update step remains quadratic (as is the case for EKF-SLAM).

VII. Simulation Results

A series of Monte-Carlo comparison studies were conducted under various conditions, in order to verify the preceding consistency analysis and to compare the performance of the proposed OC-UKF to that of the standard UKF/EKF and the OC-EKF [10], as well as the iSAM algorithm [42]. The metrics used to evaluate estimation performance are the root mean squared error (RMSE) and the average normalized (state) estimation error squared (NEES) [11]. The RMSE provides a measure of accuracy, while the NEES is a standard criterion for evaluating estimator consistency. Specifically, it is known that the NEES of an N-dimensional Gaussian random variable follows a χ^2 distribution with N d.o.f.. Therefore, if an estimator is consistent, we expect that the average NEES for the robot pose will be close to 3 for all time steps, and the average landmark NEES will be close to 2. The larger the deviations of the NEES from these values, the worse the inconsistency of the estimator. Note that when two estimators produce comparable RMSE, the one whose NEES is closer to the expected value is also the one whose estimated covariance is closer to the true covariance.[4] By studying both the RMSE and NEES of an estimator, we obtain a comprehensive picture of the estimator's performance.

A. SLAM with Range-and-Bearing Measurements

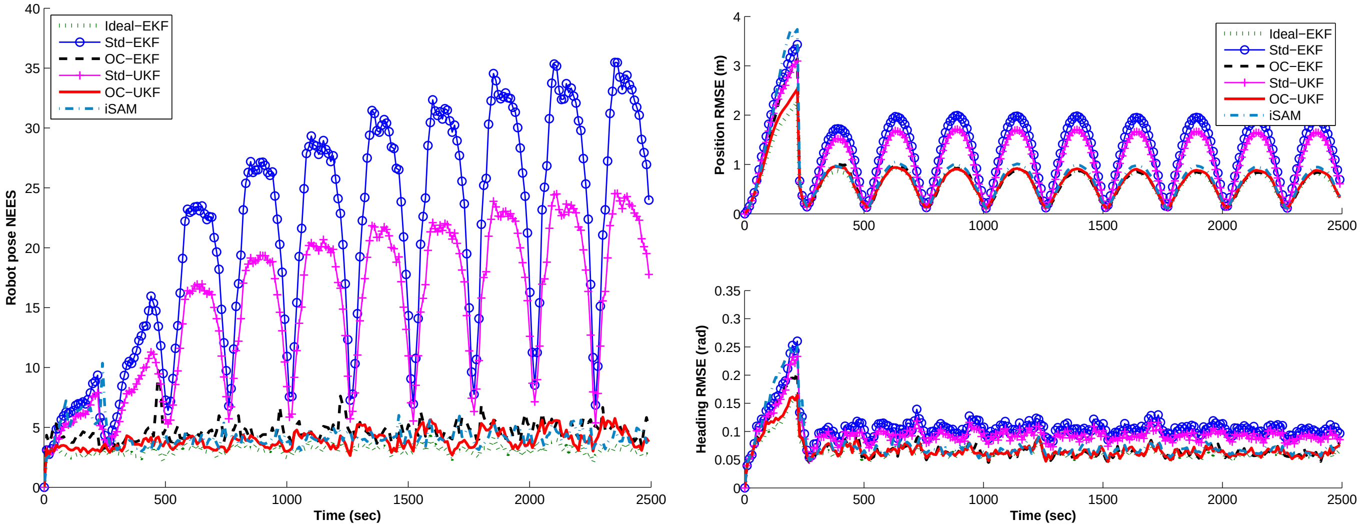
In the simulation tests presented in this section, a robot with a differential-drive model drove on a planar surface, at a constant velocity of $v = 0.25$ m/sec. The two drive wheels were equipped with encoders, which measure their revolutions and provide measurements of velocity (i.e., right and left wheel velocities, v_r and v_l, respectively), with standard deviation equal to $\sigma = 2\% v$ for each wheel. These measurements were used to obtain the linear and rotational velocity measurements for the robot, which are given by:

$$v = \frac{v_r + v_l}{2}, \qquad \omega = \frac{v_r - v_l}{a}$$

where $a = 0.5$ m is the distance between the active wheels. The robot recorded distance and bearing measurements to landmarks lying within its sensing range of 5 m. The standard deviation of the distance-measurement noise was equal to 10% of the robot-to-landmark distance, while the standard deviation of the bearing-measurement noise was set to 10 deg. It should be noted that the sensor-noise levels selected for the simulations are larger than what is typically encountered in practice. This was done on purpose in order to make the effects of inconsistency more apparent.

For the results shown here, a SLAM scenario with multiple loop closures was considered, where during each run, the robot executed 10 loops on a circular trajectory, and observed 20 landmarks in total. The reported results were averaged over 50 Monte-Carlo trials. During the test, six estimators processed

[4] It is important to stress that knowing the uncertainty of the computed estimates is often as important as the estimates themselves. An inconsistent estimator that reports covariance values smaller than the true ones can be unreliable for use in practice.



(a)

(b)

Fig. 1. [Simulation results. Range-and-bearing SLAM] Monte-Carlo results for a SLAM scenario with multiple loop closures: (a) average NEES of the robot-pose errors, and (b) average RMSE for the robot pose (position and orientation). In these plots, the dotted lines correspond to the ideal EKF, the solid lines with circles to the standard EKF, the dashed lines to the OC-EKF, the solid lines with crosses to the standard UKF, the solid lines to the OC-UKF, and the dash-dotted lines to the iSAM algorithm. Note that the RMSE of the ideal EKF, the OC-EKF, the OC-UKF and the iSAM algorithm are very close, which makes the corresponding lines difficult to distinguish.

the same data, to ensure a fair comparison.[5] The compared estimators were: (i) the ideal EKF, (ii) the standard EKF, (iii) the OC-EKF [10], (iv) the standard UKF, (v) the OC-UKF, and (vi) the iSAM algorithm [42]. Note that, as shown in [36], the performance of iSAM is very similar to (or even slightly better than) that of iSAM2 in landmark-based SLAM, which is the case considered in this work. Hence, in this test, we compared our algorithm to iSAM [42], using version 1.6 of its open-source implementation [53] with standard parameters, i.e., solving at every time step and reordering/relinearizing every 100 time steps. We also point out that, in order to ensure a fair comparison, we report the current-state estimates (instead of the final batch estimates) of the iSAM algorithm at each time step, which are computed by processing the measurements up to the current time step, without using any future measurements' information. Clearly, these incremental *causal* estimates are of more practical importance in any real-time robotic operation. Lastly, it is important to note that the ideal EKF is *not* realizable in practice since its Jacobians are evaluated at the (unknown) true values of the state. However, we included it as a benchmark in our simulations, since it has been shown to possess the *correct* observability properties and exhibit the best performance in terms of both consistency and accuracy [8]–[10], [16].

The comparative results for all the estimators are presented in Fig. 1 and Table I. Specifically, Figs. 1(a) and 1(b) show the average NEES and RMSE, respectively, over all Monte-Carlo runs for each time step for the robot pose. On the other hand,

TABLE I
[SIMULATION RESULTS. RANGE-AND-BEARING SLAM] ROBOT POSE AND LANDMARK POSITION ESTIMATION PERFORMANCE

Ideal-EKF	Std-EKF	OC-EKF	Std-UKF	OC-UKF	iSAM
Robot Position RMSE (m)					
0.6297	1.2664	0.6771	1.1002	0.6635	0.7587
Robot Heading RMSE (rad)					
0.0648	0.1070	0.0696	0.0954	0.0680	0.0760
Robot Pose NEES					
3.1284	20.6195	4.6896	14.8696	3.9305	4.2649
Landmark Position RMSE (m)					
0.6071	1.2552	0.6539	1.0890	0.6325	0.7732
Landmark Position NEES					
2.1569	19.5556	4.6150	13.7205	2.8303	10.1408

Table I presents the average values of all relevant performance metrics for the landmarks and the robot. For the landmarks, we computed the average RMSE and NEES by averaging over all Monte-Carlo runs, all landmarks, and all time steps. For the robot position and orientation RMSEs and the robot pose NEES, we averaged the corresponding quantities over all Monte-Carlo runs and all time steps.

Several interesting conclusions can be drawn from these results. Firstly, it becomes clear that the performance of the proposed 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 both RMSE (accuracy) and NEES (consistency). The observed performance gain indicates that the observability properties of the linear-regression-based

[5] In [10], the OC-EKF was shown to perform better, in terms of accuracy and consistency, than both the FEJ-EKF [8] and the robocentric mapping algorithm [52], which aims at improving the consistency of EKF-SLAM by expressing the landmarks in a robot-relative frame. Therefore, in this paper we omitted the comparison between the proposed OC-UKF and the FEJ-EKF as well as the robocentric mapping filter.

system model employed in the UKF play a key role in determining the filter consistency: When these properties differ from those of the underlying nonlinear system, which is the case for the standard EKF and UKF, the filter's consistency is negatively impacted.

A second observation is that both the OC-UKF and the OC-EKF attain slightly better performance than the iSAM algorithm, in terms of consistency and accuracy (see Fig. 1 and Table I). This can be justified by the fact that in order to reduce its processing requirements, the iSAM algorithm does *not* iteratively update the whole measurement Jacobian matrix (and thus the square-root information matrix) at every time step. Instead, it reuses partial results from the previous time steps and only updates the Jacobian matrix incrementally by appending to it new rows corresponding to the most recent measurements. However, the previously-computed parts of the Jacobian matrix can be quite inaccurate (especially right before a loop closure event or in the presence of large measurement noise). Moreover, incremental updating does not guarantee the appropriate observability properties. These factors can lead to significant estimation errors, which will propagate in time and degrade the iSAM algorithm's performance for all time steps except the ones where batch relinearization is applied. Clearly, this issue can be mitigated by performing periodic relinearization more frequently, which, however, will significantly increase the computational cost.

Lastly, the OC-UKF also outperforms the OC-EKF [10], by a smaller margin, in terms of both RMSE and NEES. It is interesting to note that the advantage of the OC-UKF over the OC-EKF is more pronounced in terms of NEES. This indicates that the OC-UKF provides a more accurate uncertainty measure (covariance) than the OC-EKF, and also implies that the filter's inconsistency primarily affects the covariance, rather than the state estimates. To further highlight this performance difference, in the next section, we also compare these algorithms in the case of bearing-only SLAM (BOSLAM), whose severe nonlinearities make the need for a better linearization scheme, such as the one offered by the OC-UKF, more evident.

B. SLAM with Bearing-only Measurements

In this BOSLAM simulation test, we employed the same simulation setup as in the preceding case, with some changes in the parameters. Specifically, the robot moved on a circular trajectory at a constant velocity of $v = 0.5$ m/sec, with wheel-velocity measurement noise standard deviation equal to $\sigma = 1\%v$, while the standard deviation of the bearing-measurement noise was set to 2 deg. Note that we doubled the robot velocity in this simulation, because a larger linear velocity increases the baseline between two consecutive time steps, leading to a more reliable triangulation-based landmark initialization [54]. Once a landmark is initialized, the inferred Jacobians for the bearing measurements to this landmark are computed in the same way as for the range-and-bearing measurements (see Section VI-B).

The comparative results[6] of the robot pose and landmark position estimation are shown in Fig. 2 and Table II. As evident, in the case of BOSLAM where the measurement nonlinearity is more significant than that of the range-and-bearing SLAM considered earlier, the standard UKF performs substantially better than the standard EKF, in terms of both consistency (NEES) and accuracy (RMSE). This performance gain is also shared by the OC-UKF over the OC-EKF. We thus see that the OC-UKF combines the benefits of the OC-EKF (i.e., correct 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.

TABLE II
[SIMULATION RESULTS. BEARING-ONLY SLAM] ROBOT POSE AND LANDMARK POSITION ESTIMATION PERFORMANCE

Ideal-EKF	Std-EKF	OC-EKF	Std-UKF	OC-UKF
Robot Position RMSE (m)				
0.0427	0.1132	0.0529	0.0707	0.0455
Robot Heading RMSE (rad)				
0.0045	0.0130	0.0055	0.0075	0.0043
Robot Pose NEES				
2.6054	12.6715	4.4730	4.8453	2.6917
Landmark Position RMSE (m)				
0.1066	0.1770	0.1305	0.1630	0.1471
Landmark Position NEES				
1.8964	12.7627	12.6085	6.1927	4.3216

VIII. EXPERIMENTAL RESULTS

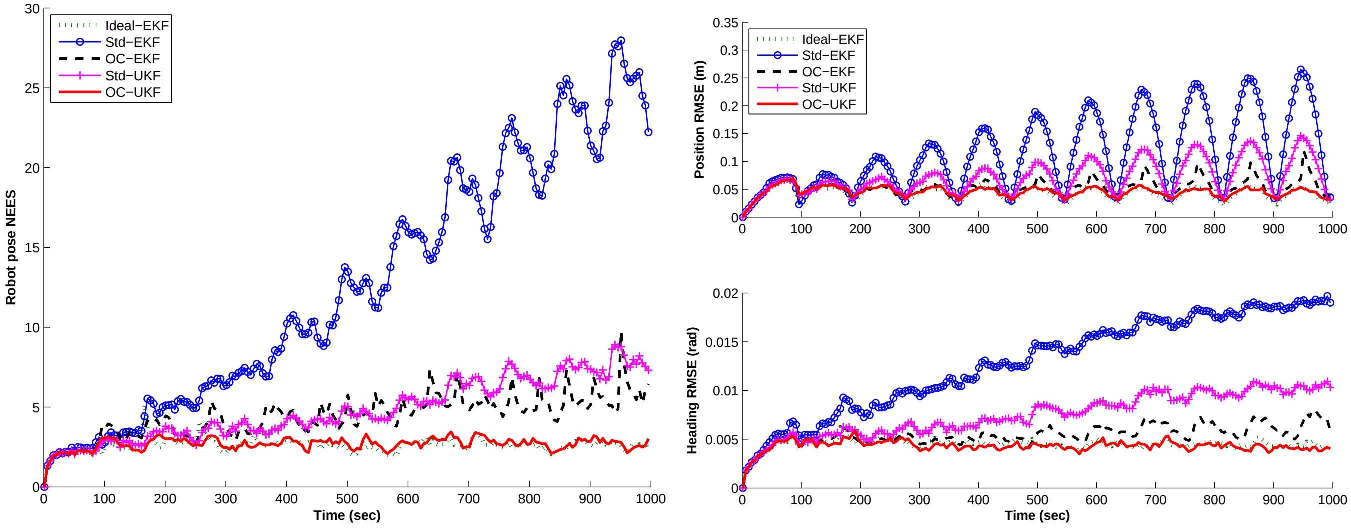
To further test the proposed OC-UKF SLAM algorithm, we also conducted real-world experiments in both indoor and outdoor environments. These tests also allow us to examine the algorithm's runtime, as compared to the OC-EKF and the state-of-the-art iSAM algorithm. All the timing results presented in this section were obtained on a Mac laptop with an Intel i5 processor at 2.53 GHz, and 4GB of RAM.

A. Indoor Environment

We first present the results of the indoor experiment conducted in an office building. The robot was commanded to perform 11 loops around a square with sides approximately equal to 20 m (see Fig. 3). This trajectory was selected since repeated re-observation of the same landmarks tends to make the effects of inconsistency more apparent, and facilitates discerning the performance of the various estimators. A Pioneer robot equipped with a SICK LMS200 laser range-finder and wheel encoders was used in this experiment. From the laser-range data, corner features were extracted and used as landmarks, while the wheel encoders provided the linear and rotational velocity measurements. In particular, this dataset was recorded over about 40 minutes, and contains 23425 robot poses and 63 landmarks with 11392 measurements to them.

Since no ground truth for the robot pose could be obtained using external sensors (e.g., overhead cameras) in this experiment, we obtained a reference trajectory, treated as ground

[6] Since the current implementation of iSAM [53] does not include the bearing-only case, in this test we omit the comparison of the OC-UKF to the iSAM algorithm.



(a)

(b)

Fig. 2. [Simulation results. Bearing-only SLAM] Monte-Carlo results for a SLAM scenario with multiple loop closures: (a) average NEES of the robot-pose errors, and (b) average RMSE for the robot pose (position and orientation). In these plots, the dotted lines correspond to the ideal EKF, the solid lines with circles to the standard EKF, the dashed lines to the OC-EKF, the solid lines with crosses to the standard UKF, and the solid lines to the OC-UKF. Note that the RMSE of the ideal EKF and the OC-UKF are almost identical, which makes the corresponding lines difficult to distinguish.

truth, by utilizing the known map of the area where the experiment took place. Specifically, the exact locations of 20 corners were known from the blueprints of the building. Measurements to these corners, as well as all other measurements obtained by the robot (including those to corners whose locations were not known *a priori*), were processed offline using a batch maximum a posteriori (MAP) estimator [55] to obtain an accurate estimate of the entire trajectory. This estimate, as well as the locations of the known corners, are shown in Fig. 3. This constitutes the ground truth against which the performance of the following five estimators was compared: (i) the standard EKF, (ii) the OC-EKF, (iii) the standard UKF, (iv) the OC-UKF, and (v) iSAM. Clearly, due to the way the ground truth is computed, the estimation errors are expected to have some correlation to the errors in the ground truth. However, since these correlations are the same for all estimators, we can still have a fair comparison of their relative performance.

The comparative results for all estimators are presented in Figs. 4(a) and 4(b), while Table III shows the averaged NEES and RMSE of the robot pose and landmark position, respectively. We point out that during the experiment the robot detected a number of landmarks that were *not* included in the set of 20 known corners (e.g., movable objects such as furniture). Since no ground truth was available for these objects, we only used the 20 known corners for computing the landmarks' error statistics. From the experimental results, it becomes evident that the OC-UKF outperforms both the standard EKF and UKF, and also achieves better accuracy than the OC-EKF. This agrees with the simulation results presented in the preceding section. It should be noted that the reported NEES in Fig. 4(a) was computed only from a single run (i.e., this is not an average over many Monte-Carlo runs as in the simulations). To evaluate an estimator's consistency, the average NEES over many Monte-Carlo runs is a suitable metric, while the NEES values in a single experiment do not dictate which estimator is consistent or not. Regardless, we show these results mainly to demonstrate the large difference in performance between the OC-EKF/UKF and the standard EKF/UKF. These experimental results, along with those from the simulations, further support our conjecture that the mismatch in the dimension of the unobservable subspace between

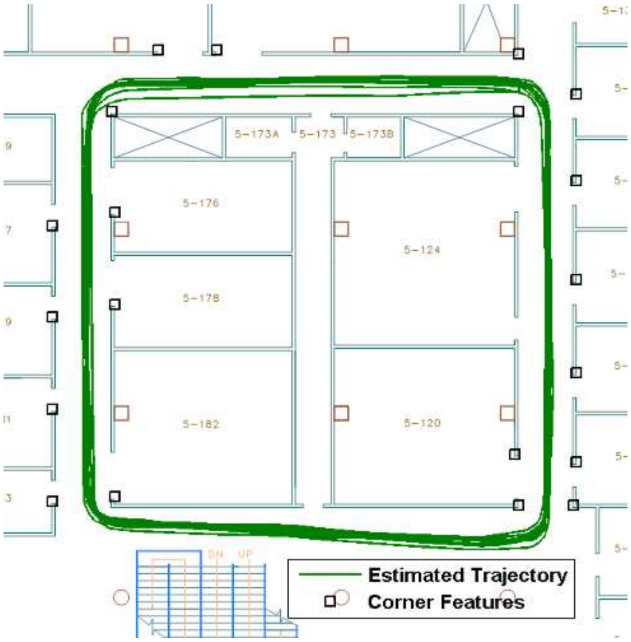
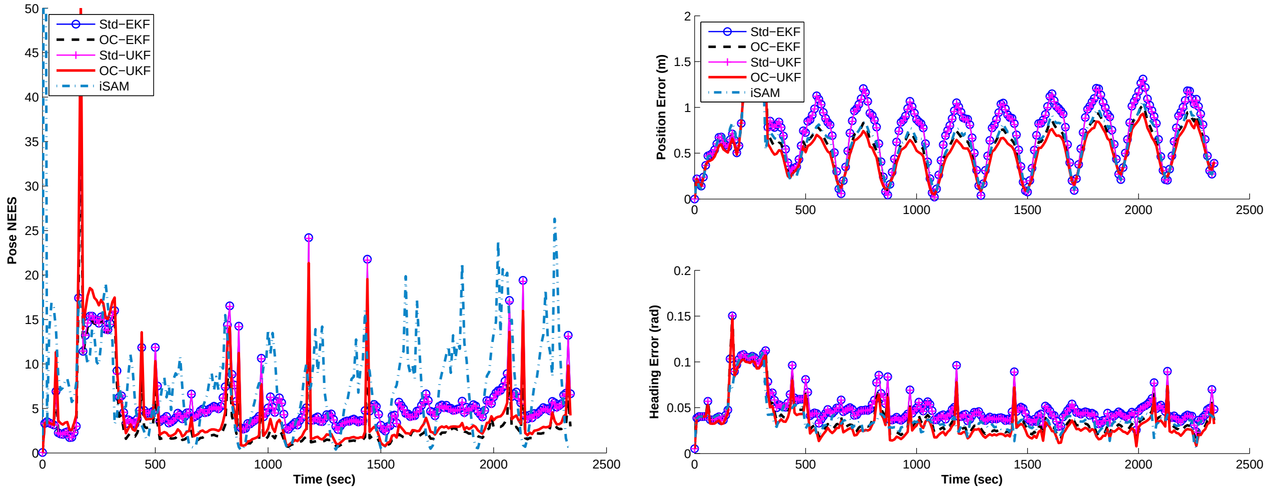


Fig. 3. [Indoor Experiment] The batch maximum-a-posteriori (MAP) estimate of the robot trajectory during the indoor experiment (solid line), overlaid on the blueprint of the building. The boxes (□) denote the corners whose exact locations were known from the building's blueprints. The batch MAP estimates of the robot poses and the known corners were used as ground truth for computing the NEES and RMSE values shown in Table III and Fig. 4.



(a)

(b)

Fig. 4. [Indoor Experiment] Experimental results: (a) NEES of the robot-pose errors, and (b) estimation errors of the robot pose (position and orientation). In these plots, the solid lines with circles correspond to the standard EKF, the dashed lines to the OC-EKF, the solid lines with crosses to the standard UKF, the solid lines to the OC-UKF, and the dash-dotted lines to the iSAM algorithm. Note that the NEES and estimation error values of the standard EKF and the standard UKF are almost identical, and the estimation errors of the OC-UKF, the OC-EKF and the iSAM algorithm are also very close to each other, which makes the corresponding lines difficult to distinguish.

TABLE III
[Indoor Experiment] Robot Pose and Landmark Position Estimation Performance and Runtime

Std-EKF	OC-EKF	Std-UKF	OC-UKF	iSAM
		Robot Position RMSE (m)		
0.7323	0.5896	0.7268	0.5384	0.6108
		Robot Heading RMSE (rad)		
0.0512	0.0392	0.0508	0.0349	0.0388
		Robot Pose NEES		
6.0939	3.4575	6.0307	4.5442	9.1270
		Landmark Position RMSE (m)		
0.9929	0.8438	0.9894	0.8183	0.6528
		Landmark Position NEES		
7.3180	6.0354	7.2928	7.0123	9.6627
		Total CPU Execution Time (sec)		
304.761	304.251	306.689	307.930	350.379

the statistically-linearized SLAM system and the underlying nonlinear system is a fundamental cause of filter inconsistency.

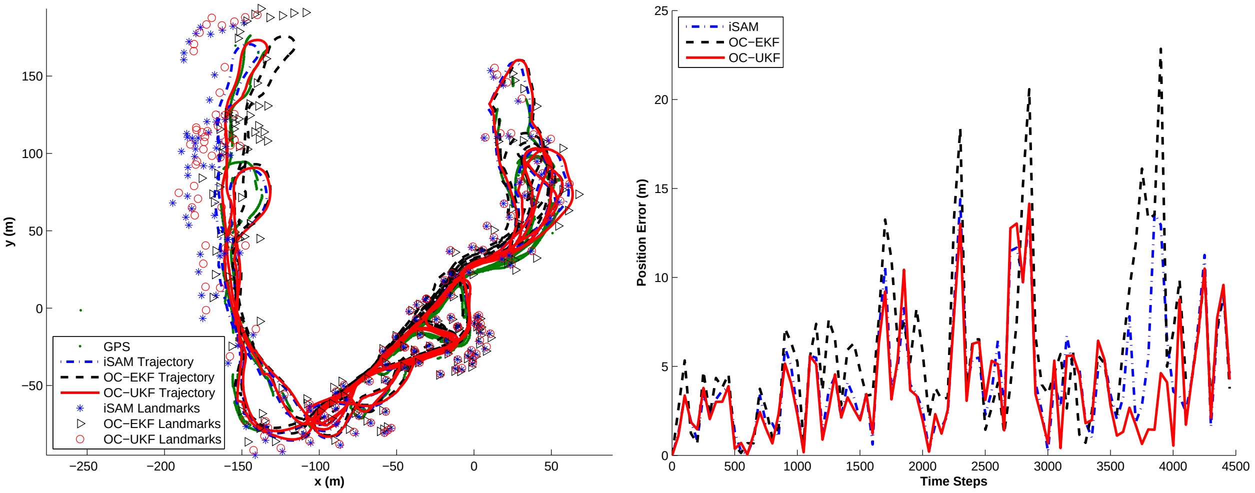
As evident from Fig. 4(b) and Table III, the OC-EKF/UKF achieve similar accuracy to, and better consistency than, the iSAM algorithm. As mentioned in the previous section, one possible explanation for this is that the iSAM algorithm does not iteratively update the whole measurement Jacobian at each time step, which may incur large linearization and thus estimation errors. Inaccuracies in the measurement Jacobian propagate into the covariance estimated by the iSAM algorithm, which results in significantly higher NEES values as compared to the OC-EKF/UKF. Interestingly, as seen from Table III, the OC-UKF has a lower computational cost than iSAM in this experiment, although all the algorithms attain faster-than-real-time performance. This can be justified by the fact that the computational cost of the iSAM algorithm increases as the robot trajectory grows. Moreover, the 11 loop-closing events occurring along the robot trajectory in this experiment significantly increase fill-in in the square-root information matrix and thus the computational complexity for solving the system.

At this point we should note that in this indoor experiment (as well as the outdoor experiment presented in the next section) the measurement correspondences were known. If not, then to solve the data association problem, the iSAM algorithm would need to recover marginal covariances, which will significantly increase its processing requirements [42]. By contrast, since the covariance matrix is maintained in the OC-EKF/UKF, the marginal covariances are immediately available and hence the maximum-likelihood data association incurs minimal overhead.

Lastly, it is very important to observe from Table III that the two UKFs (i.e., the standard UKF and the proposed OC-UKF) have similar timing performance as the two EKFs (i.e, the standard EKF and the OC-EKF). This is attributed to the proposed sampling strategy (see Lemma IV.1), which results in the UKF having computational complexity of the same order as that of the EKF.

B. Outdoor Environment

To further examine the performance of the proposed OC-UKF, we tested our algorithm on a publicly available SLAM dataset, the Sydney Victoria Park dataset. The experimental platform was a 4-wheeled vehicle equipped with a kinematic GPS, a laser sensor, and wheel encoders. The GPS system was used to provide ground truth for the robot position. Wheel encoders were used to provide odometry measurements, and



(a)

(b)

Fig. 5. [Outdoor Experiment] Experimental results: (a) robot trajectory and landmark estimates as compared to the GPS data, and (b) robot position estimation errors. Note that, since in this test the GPS satellite signals were not always available, we computed the estimation errors only when GPS was available. In these plots, the dash-dotted lines and stars correspond to the iSAM estimates of the trajectory and the landmarks, respectively, the dashed lines and triangles to the OC-EKF, and the solid lines and circles to the OC-UKF, while the dots denote the sparse GPS data points.

TABLE IV
[OUTDOOR EXPERIMENT] ESTIMATION ACCURACY AND RUNTIME

iSAM	OC-EKF	OC-UKF
	Robot Position RMSE (m)	
4.2111	5.9069	3.8084
	Total CPU Execution Time (sec)	
31.5482	35.6811	34.6138

propagation was carried out using the Ackerman model. In this particular application, since the most common features in the environment were trees, the profiles of trees were extracted from the laser data, and the centers of the trunks were then used as the point landmarks. It should be pointed out that in this test, to ensure a fair comparison with the iSAM algorithm, we employed the preprocessed dataset which is also available in the iSAM package [53]. This preprocessed dataset contains 6969 robot poses and 151 landmarks with 3640 measurements, recorded over 26 minutes.

Since the OC-EKF and the OC-UKF were already shown in the preceding simulations and experiment to perform significantly better, in terms of accuracy and consistency, than the standard EKF and UKF, in this test, we omitted the comparison to the two latter filters for clarity of presentation. Instead, we focus on the accuracy comparison of the OC-UKF with the OC-EKF and the iSAM algorithm. In this experiment, true landmark positions and true robot orientations were not available. We hence only compared the position-estimation performance of the three approaches (i.e., the OC-EKF, the OC-UKF, and the iSAM algorithm). Note also that, as mentioned in Section VIII-A, the NEES computed from a single experimental run is not well-suited for analyzing the consistency of the estimators, and thus we hereafter focus on the comparison of accuracy and processing requirements. Specifically, Fig. 5(a) depicts the trajectory and landmark estimates produced by the three estimators as compared to the GPS ground truth, while Fig. 5(b) shows the corresponding estimation errors of the robot position. Table IV shows the average estimation errors (i.e., RMSE) of robot position as well as the total CPU runtime for the three estimators compared. Clearly, the OC-UKF achieves better accuracy than both the OC-EKF and iSAM, while incurring comparable computational cost. In particular, the OC-UKF attains 36% and 10% reduction in robot position estimation errors as compared to the OC-EKF and iSAM, while at 3% lower and only 10% higher computational cost, respectively.[7] We repeat that the timing result for iSAM does not include the runtime of computing marginal covariances for data association. These results agree with what we have observed in the indoor experiment presented in Section VIII-A.

IX. CONCLUSIONS AND FUTURE WORK

This work focuses on UKF-based SLAM, and particularly on the issues of computational complexity and filter inconsistency. The first contribution of this work is the formulation of a novel UKF-based SLAM algorithm that has computational complexity of the same order as that of EKF-based SLAM. In particular, we have proposed a new sampling scheme in which the unscented transformation employed by the UKF is only applied to the subset of states that appear in the nonlinear process and measurement models, instead of the entire state. Thus, by adopting this new sampling scheme, the UKF-based SLAM requires computing the square root of small, constant-size matrices, which leads to computational complexity *linear* during propagation, and *quadratic* during update. Furthermore, we

[7] Note that in this experiment, the OC-UKF has lower cost than the OC-EKF, primarily because the Mahalanobis-distance test [11] in the OC-UKF rejects more outlier measurements than that in the OC-EKF.

have shown that a mismatch between the observability properties of the linear-regression-based system model employed by the UKF, and those of the underlying nonlinear SLAM system, causes inconsistency. To address this issue, we have introduced a novel observability-constrained (OC)-UKF, which ensures that the UKF system model has an unobservable subspace of appropriate dimensions, by enforcing observability constraints on the filter's *inferred* Jacobians. Through extensive Monte-Carlo simulations and real-world experiments, the OC-UKF is shown to achieve comparable or better performance, in terms of consistency, accuracy and computational complexity, to other state-of-the-art SLAM algorithms such as the OC-EKF and iSAM.

In this work, we have focused on 2D SLAM. However, the proposed approach is also applicable to the case of robot localization in 3D using inertial sensors. The details of the application of the proposed OC-UKF to 3D SLAM will be the focus of our future work.

Appendix A
Proof of Lemma IV.1

We start with the linearization error in the linear regression:

$$\mathbf{e}(\mathbf{x}) = \mathbf{y} - (\mathbf{A}\mathbf{x} + \mathbf{b}) = \ \mathbf{y} - \mathbf{A}_1\mathbf{x}_1 - \mathbf{A}_2\mathbf{x}_2 - \mathbf{b} \qquad (66)$$

Substituting (66) in the expression of the expected value of the squared linearization error (1), the cost function we seek to minimize becomes:

$$\int_{-\infty}^{+\infty} ||\mathbf{y} - \mathbf{A}_1\mathbf{x}_1 - \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}||^2 p(\mathbf{x}) d\mathbf{x} \ = \\ \iint_{-\infty}^{+\infty} ||\mathbf{y} - \mathbf{A}_1\mathbf{x}_1 - \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}||^2 p(\mathbf{x}_2|\mathbf{x}_1) p(\mathbf{x}_1) d\mathbf{x}_1 d\mathbf{x}_2 \qquad (67)$$

where we have employed the notation $||\boldsymbol{\alpha}||^2 \triangleq \boldsymbol{\alpha}^T\boldsymbol{\alpha}$, and the property $p(\mathbf{x}) = p(\mathbf{x}_1, \mathbf{x}_2) = p(\mathbf{x}_2|\mathbf{x}_1)p(\mathbf{x}_1)$. Now using the standard sample-based approximation, $p(\mathbf{x}_1) \simeq \sum_{i=0}^{r} w_i \delta(\mathbf{x}_1 - \mathcal{X}_{1_i})$, where the samples are selected to match the mean and covariance of $p(\mathbf{x}_1)$ [see (2) and (3)], we rewrite the cost function (67) as follows:

$$\begin{aligned} c :=& \sum_{i=0}^{r} w_i \int_{-\infty}^{+\infty} ||\mathcal{Y}_i - \mathbf{A}_1\mathcal{X}_{1_i} - \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}||^2 p(\mathbf{x}_2|\mathbf{x}_1 = \mathcal{X}_{1_i}) d\mathbf{x}_2 \\ =& \sum_{i=0}^{r} w_i \mathbb{E}\left(||\mathcal{Y}_i - \mathbf{A}_1\mathcal{X}_{1_i} - \mathbf{A}_2\mathbf{x}_2 - \mathbf{b}||^2\right) \\ =& \sum_{i=0}^{r} w_i \Big[(\mathcal{Y}_i - \mathbf{A}_1\mathcal{X}_{1_i} - \mathbf{b})^T (\mathcal{Y}_i - \mathbf{A}_1\mathcal{X}_{1_i} - \mathbf{b}) \ - \\ & 2\left(\mathcal{Y}_i - \mathbf{A}_1\mathcal{X}_{1_i} - \mathbf{b}\right)^T \mathbf{A}_2\mathbb{E}(\mathbf{x}_2)\Big] + \mathrm{tr}\left[\mathbf{A}_2\mathbb{E}(\mathbf{x}_2\mathbf{x}_2^T)\mathbf{A}_2^T\right] \qquad (68) \end{aligned}$$

where have used the following identity:

$$\mathbb{E}(\mathbf{x}_2^T\mathbf{A}_2^T\mathbf{A}_2\mathbf{x}_2) = \mathrm{tr}\left[\mathbf{A}_2\mathbb{E}(\mathbf{x}_2\mathbf{x}_2^T)\mathbf{A}_2^T\right] \qquad (69)$$

Note that the expectation operator $\mathbb{E}(\cdot)$ is with respect to the pdf $p(\mathbf{x}_2|\mathbf{x}_1 = \mathcal{X}_{1_i})$. For the Gaussian case, this pdf can be expressed analytically as follows:

$$p(\mathbf{x}_2|\mathbf{x}_1 = \mathcal{X}_{1_i}) \ = \mathcal{N}\left(\hat{\mathcal{X}}_{2_i}, \bar{\mathbf{P}}_{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_2}\right) = \qquad (70) \\ \mathcal{N}\left(\underbrace{\hat{\mathbf{x}}_2 + \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}\left(\mathcal{X}_{1_i} - \hat{\mathbf{x}}_1\right)}_{\hat{\mathcal{X}}_{2_i}}, \underbrace{\mathbf{P}_{\mathbf{x}_2\mathbf{x}_2} - \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_2}}_{\bar{\mathbf{P}}_{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_2}}\right)$$

Based on (70), we have:

$$\mathbb{E}\left(\mathbf{x}_2\mathbf{x}_2^T\right) = \mathbb{E}\left(\mathbf{x}_2\right)\mathbb{E}\left(\mathbf{x}_2\right)^T + \bar{\mathbf{P}}_{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_2} = \hat{\mathcal{X}}_{2_i}\hat{\mathcal{X}}_{2_i}^T + \bar{\mathbf{P}}_{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_2} \qquad (71)$$

Substituting $\mathbb{E}(\mathbf{x}_2) \triangleq \hat{\mathcal{X}}_{2_i}$ from (70) and $\mathbb{E}(\mathbf{x}_2\mathbf{x}_2^T)$ from (71) in (68) and simplifying it, we obtain:

$$c = \sum_{i=0}^{r} w_i ||\mathcal{Y}_i - \mathbf{A}_1\mathcal{X}_{1_i} - \mathbf{A}_2\hat{\mathcal{X}}_{2_i} - \mathbf{b}||^2 + \mathrm{tr}\left(\mathbf{A}_2\bar{\mathbf{P}}_{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_2}\mathbf{A}_2^T\right) \qquad (72)$$

Our goal is to minimize the cost function in (72) with respect to $\mathbf{b}$ and $\mathbf{A} \triangleq \begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}$. To do so, we first compute the optimal solution for $\mathbf{b}$, by setting the derivative of (72) with respect to $\mathbf{b}$ to zero. This yields:

$$\frac{\partial c}{\partial \mathbf{b}} = -2\sum_{i=0}^{r} w_i \left(\mathcal{Y}_i - \mathbf{A}_1\mathcal{X}_{1_i} - \mathbf{A}_2\hat{\mathcal{X}}_{2_i} - \mathbf{b}\right)^T = \mathbf{0} \ \Rightarrow \\ \mathbf{b} = \sum_{i=0}^{r} w_i\mathcal{Y}_i - \mathbf{A}_1\sum_{i=0}^{r} w_i\mathcal{X}_{1_i} - \mathbf{A}_2\sum_{i=0}^{r} w_i\hat{\mathcal{X}}_{2_i} = \bar{\mathbf{y}} - \mathbf{A}_1\hat{\mathbf{x}}_1 - \mathbf{A}_2\hat{\mathbf{x}}_2 \qquad (73)$$

where for the last step, we have used (6), the equality $\hat{\mathbf{x}}_1 = \sum_{i=0}^{r} w_i\mathcal{X}_{1_i}$, and the identity $\hat{\mathbf{x}}_2 = \sum_{i=0}^{r} w_i\hat{\mathcal{X}}_{2_i}$, which stems from (70).

Substituting (73) in (72), we have:

$$c' = \sum_{i=0}^{r} w_i ||\tilde{\mathcal{Y}}_i - \mathbf{A}_1\tilde{\mathcal{X}}_{1_i} - \mathbf{A}_2\tilde{\mathcal{X}}_{2_i}||^2 + \mathrm{tr}\left(\mathbf{A}_2\bar{\mathbf{P}}_{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_2}\mathbf{A}_2^T\right) \qquad (74)$$

where

$$\tilde{\mathcal{X}}_{1_i} \triangleq \mathcal{X}_{1_i} - \hat{\mathbf{x}}_1 \ , \quad \tilde{\mathcal{Y}}_i \triangleq \mathcal{Y}_i - \bar{\mathbf{y}} \qquad (75)$$

$$\tilde{\mathcal{X}}_{2_i} \triangleq \hat{\mathcal{X}}_{2_i} - \hat{\mathbf{x}}_2 = \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}\tilde{\mathcal{X}}_{1_i} \qquad (76)$$

Note that (70) was used in (76). Taking derivatives of the cost function in (74) with respect to $\mathbf{A}_1$ and $\mathbf{A}_2$, and setting them equal to zero, we obtain:

$$\frac{\partial c'}{\partial \mathbf{A}_1} = -2\sum_{i=0}^{r} w_i \left(\tilde{\mathcal{Y}}_i - \mathbf{A}_1\tilde{\mathcal{X}}_{1_i} - \mathbf{A}_2\tilde{\mathcal{X}}_{2_i}\right)\tilde{\mathcal{X}}_{1_i}^T = \mathbf{0} \qquad (77)$$

$$\frac{\partial c'}{\partial \mathbf{A}_2} = -2\sum_{i=0}^{r} w_i \left(\tilde{\mathcal{Y}}_i - \mathbf{A}_1\tilde{\mathcal{X}}_{1_i} - \mathbf{A}_2\tilde{\mathcal{X}}_{2_i}\right)\tilde{\mathcal{X}}_{2_i}^T + 2\mathbf{A}_2\bar{\mathbf{P}}_{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_2} = \mathbf{0} \qquad (78)$$

At this point, we use the fact that, due to the selection of the sigma points, we have:

$$\sum_{i=0}^{r} w_i\tilde{\mathcal{Y}}_i\tilde{\mathcal{X}}_{1_i}^T = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_1} \ , \quad \sum_{i=0}^{r} w_i\tilde{\mathcal{X}}_{1_i}\tilde{\mathcal{X}}_{1_i}^T = \mathbf{P}_{\mathbf{x}_1\mathbf{x}_1} \qquad (79)$$

$$\sum_{i=0}^{r} w_i\tilde{\mathcal{X}}_{2_i}\tilde{\mathcal{X}}_{1_i}^T = \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}\sum_{i=0}^{r} w_i\tilde{\mathcal{X}}_{1_i}\tilde{\mathcal{X}}_{1_i}^T = \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1} \qquad (80)$$

$$\sum_{i=0}^{r} w_i\tilde{\mathcal{Y}}_i\tilde{\mathcal{X}}_{2_i}^T = \sum_{i=0}^{r} w_i\tilde{\mathcal{Y}}_i\tilde{\mathcal{X}}_{1_i}^T\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} \qquad (81)$$

$$\sum_{i=0}^{r} w_i\tilde{\mathcal{X}}_{2_i}\tilde{\mathcal{X}}_{2_i}^T = \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}\left(\sum_{i=0}^{r} w_i\tilde{\mathcal{X}}_{1_i}\tilde{\mathcal{X}}_{1_i}^T\right)\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} \\ = \mathbf{P}_{\mathbf{x}_2\mathbf{x}_1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} \qquad (82)$$

where (76) was used for deriving these relations. Substituting the above results in (77) and (78) yields:

$$\bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_1} - \mathbf{A}_1\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1} - \mathbf{A}_2\mathbf{P}_{\mathbf{x}_2\mathbf{x}_1} = \mathbf{0} \qquad (83)$$

$$\bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} - \mathbf{A}_1\mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} - \mathbf{A}_2\mathbf{P}_{\mathbf{x}_2\mathbf{x}_1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_2} - \mathbf{A}_2\bar{\mathbf{P}}_{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_2} = \mathbf{0} \qquad (84)$$

It is easy to verify (e.g., by substitution) that the solution to the system of equations (83)-(84) is $\mathbf{A}_1 = \bar{\mathbf{P}}_{\mathbf{y}\mathbf{x}_1}\mathbf{P}_{\mathbf{x}_1\mathbf{x}_1}^{-1}$ and $\mathbf{A}_2 = \mathbf{0}$. This completes the proof.

APPENDIX B
PROOF OF LEMMA VI.1

Using (65), we write the equality constraint on $\mathbf{A}_1$ (62) as $\mathbf{A}_1\mathbf{U} = \mathbf{0}$. This equation states that the rows of $\mathbf{A}_1$ 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 $\mathbf{A}_1$ as:

$$\mathbf{A}_1 = \mathbf{B}\mathbf{L} \tag{85}$$

where $\mathbf{B}$ is an $m \times 2$ unknown 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 (64). Substituting (85) in the original problem formulation [see (4) and (61)], we obtain:

$$\min_{\mathbf{B},\mathbf{b}} \sum_{i=0}^{10} w_i \left[\mathcal{Z}_i - (\mathbf{B}\mathcal{X}'_i + \mathbf{b})\right]^T \left[\mathcal{Z}_i - (\mathbf{B}\mathcal{X}'_i + \mathbf{b})\right] \tag{86}$$

where we have defined $\mathcal{X}'_i \triangleq \mathbf{L}\mathcal{X}_i$, $i = 0, \ldots, 10$. This is an unconstrained minimization problem with respect to the design variables $\mathbf{B}$ and $\mathbf{b}$, and has exactly the same structure as that in (4). Thus, by analogy, the optimal solution for $\mathbf{B}$ is computed by [see (5)]:

$$\mathbf{B} = \bar{\mathbf{P}}_{\mathbf{z}\boldsymbol{\ell}} \bar{\mathbf{P}}_{\boldsymbol{\ell\ell}}^{-1} \tag{87}$$

where

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

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

By combining these two identities with those of (87) and (85), we obtain the optimal solution of $\mathbf{A}_1$ [see (63)].

REFERENCES

[1] P. Newman, "On the structure and solution of the simultaneous localization and mapping problem," Ph.D. dissertation, University of Sydney, 1999.

[2] S. B. Williams, P. Newman, M. Dissanayake, and H. Durrant-Whyte, "Autonomous underwater simultaneous localisation and map building," in *Proc. of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, Apr. 24–28, 2000, pp. 1793–1798.

[3] J. H. Kim and S. Sukkarieh, "Airborne simultaneous localisation and map building," in *Proc. of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sep. 14–19, 2003, pp. 406–411.

[4] 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 21–26, 2001, pp. 4238–4243.

[5] 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. 9–15, 2006, pp. 3562–3568.

[6] 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 15–19, 2006, pp. 412–417.

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

[8] 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 19–23, 2008, pp. 473–479.

[9] ——, "A first-estimates Jacobian EKF for improving SLAM consistency," in *Proc. of the 11th International Symposium on Experimental Robotics*, Athens, Greece, Jul. 14–17, 2008.

[10] ——, "Observability-based rules for designing consistent EKF SLAM estimators," *International Journal of Robotics Research*, vol. 29, no. 5, pp. 502–528, Apr. 2010.

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

[12] 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.

[13] 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*, Aug. 2–6, 2005, pp. 3427–3432.

[14] 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 19–23, 2008, pp. 3710–3716.

[15] ——, "An $O(N^2)$ square root unscented Kalman filter for visual simultaneous localization and mapping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 1251–1263, Jul. 2009.

[16] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "On the complexity and consistency of UKF-based SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 12–17, 2009, pp. 4401–4408.

[17] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1987.

[18] S. Thrun, "Robotic mapping: A survey," in *Exploring artificial intelligence in the new millennium*, G. Lakemeyer and B. Nebel, Eds. San Francisco, CA: Morgan Kaufmann Inc., 2003, pp. 1–35.

[19] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. Cambridge, MA: The MIT Press, 2005.

[20] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[21] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[22] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, Dec. 2006.

[23] K. Konolige, G. Grisetti, R. Kummerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2D mapping," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 18–22, 2010, pp. 22–29.

[24] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9–13, 2011, pp. 3607–3613.

[25] G. Grisetti, C. Stachniss, and W. Burgard, "Nonlinear constraint network optimization for efficient map learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 428–439, Sep. 2009.

[26] G. Grisetti, R. Kummerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2D and 3D mapping," in *Proc. of the IEEE International Conference on Robotics and Automation*, Anchorage, AK, May 3–8, 2010, pp. 273–278.

[27] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[28] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" *Image and Vision Computing*, vol. 30, no. 2, pp. 65–77, Feb. 2012.

[29] D. Brown, "A solution to the general problem of multiple station analytical stereo triangulation," Patrick Air Force Base, Florida, RCA-MTP Data Reduction Technical Report, no. 43, 1958.

[30] ——, "Results in geodetic photogrammetry I: The precise determination of the location of Bermuda from photogrametria observations of flares ejected from Juno 11," Patrick Air Force Base, Florida, RCA-MTP Data Reduction Technical Report, no. 54, 1959.

[31] ——, "The bundle adjustment - progress and prospects," *International Archives of Photogrammetry*, vol. 21, no. 3, pp. 1–33, 1976.

[32] S. I. Granshaw, “Bundle adjustment methods in engineering photogrammetry,” *The Photogrammetric Record*, vol. 10, no. 56, pp. 181–207, Oct. 1980.

[33] R. Szeliski and S. B. Kang, “Recovering 3D shape and motion from image streams using nonlinear least squares,” *Journal of Visual Communication and Image Representation*, vol. 5, no. 1, pp. 10–28, Mar. 1994.

[34] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment – a modern synthesis,” *Lecture Notes in Computer Science*, vol. 1883, pp. 298–375, Jan. 2000.

[35] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[36] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping using the Bayes tree,” *International Journal of Robotics Research*, vol. 31, pp. 217–236, Feb. 2012.

[37] G. Sibley, L. Matthies, and G. Sukhatme, “Sliding window filter with application to planetary landing,” *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608, Sep./Oct. 2010.

[38] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, “An observability constrained sliding window filter for SLAM,” in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, Sep. 25–30, 2011, pp. 65–72.

[39] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” in *Proc. of the IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nara, Japan, Nov. 13–16, 2007.

[40] K. Konolige and M. Agrawal, “FrameSLAM: From bundle adjustment to real-time visual mapping,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1066–1077, Oct. 2008.

[41] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, “View-based maps,” *International Journal of Robotics Research*, vol. 29, no. 8, pp. 941–957, Jul. 2010.

[42] M. Kaess, A. Ranganathan, and F. Dellaert, “iSAM: Incremental smoothing and mapping,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[43] M. Kaess, V. Ila, R. Roberts, and F. Dellaert, “The Bayes tree: An algorithmic foundation for probabilistic robot mapping,” in *Proc. of the International Workshop on the Algorithmic Foundations of Robotics*, Singapore, Dec. 2010, pp. 157–173.

[44] H. Strasdat, J. Montiel, and A. Davison, “Real-time monocular SLAM: Why filter?” in *Proc. of the IEEE International Conference on Robotics and Automation*, Anchorage, AK, May 3–8, 2010, pp. 2657–2664.

[45] J. Langelaan and S. Rock, “Passive GPS-free navigation for small UAVs,” in *Proc. of the IEEE Aerospace Conference*, Big Sky, MT, Mar. 5–12, 2005, pp. 1–9.

[46] 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 International Symposium on Visual Computing*, Lake Tahoe, NV, Nov. 6–8, 2006, pp. 276–285.

[47] 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. 18–22, 2005, pp. 323–328.

[48] 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.

[49] K. Lee, W. Wijesoma, and J. Guzman, “On the observability and observability analysis of SLAM,” in *Proc. of the IEEE/RSJ Intenational Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 9–15, 2006, pp. 3569–3574.

[50] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Academic Press, 1979, vol. 1.

[51] G. H. Golub and C. F. V. Loan, *Matrix Computations*. The Johns Hopkins University Press, 1996.

[52] J. A. 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, Jul. 5–7, 2004, pp. 1244–1249.

[53] M. Kaess, H. Johannsson, D. Rosen, and J. Leonard, “iSAM: Incremental smoothing and mapping,” https://svn.csail.mit.edu/isam, accessed Aug. 31, 2012.

[54] T. Bailey, “Constrained initialisation for bearing-only SLAM,” in *Proc. of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sep. 14–19, 2003, pp. 1966–1971.

[55] S. Kay, *Fundamentals of Statistical Signal Processing, Vol. I - Estimation Theory*. Prentice Hall, 1993.

Guoquan P. Huang received the B.Eng. in automation (electrical engineering) from the University of Science and Technology Beijing, China, in 2002 and the M.Sc. and Ph.D. degrees in computer science from the University of Minnesota, Twin Cities, in 2009 and 2012, respectively.

He is currently a Postdoctoral Associate in the Computer Science and Artificial Intelligence Laboratory (CSAIL) at the Massachusetts Institute of Technology (MIT). From 2003 to 2005, he was a Research Assistant with the Department of Electrical Engineering at the Hong Kong Polytechnic University, Hong Kong. His research interests lie in robotics and computer vision, with special emphasis on probabilistic perception, estimation, and control of autonomous ground, aerial, and underwater vehicles.

Dr. Huang received the 2006 Academic Excellence Fellowship from the University of Minnesota, the 2011 Chinese Government Award for Outstanding Self-Financed Students Abroad, and was the Finalist for the 2009 Best Paper Award from the Robotics: Science and Systems (RSS) Conference.

Anastasios I. Mourikis received the Diploma in electrical engineering from the University of Patras, Patras, Greece, in 2003, and the Ph.D. degree in computer science from the University of Minnesota, Twin Cities, in 2008.

He is currently an Assistant Professor in the Department of Electrical Engineering at the University of California, Riverside (UCR). His research interests lie in the areas of vision-aided inertial navigation, resource-adaptive estimation algorithms, distributed estimation in mobile sensor networks, simultaneous localization and mapping, and structure from motion.

Dr. Mourikis has been awarded the 2013 National Science Foundation (NSF) CAREER Award, the 2011 Hellman Faculty Fellowship Award, and is a co-recipient of the IEEE Transactions on Robotics 2009 Best Paper Award (King-Sun Fu Memorial Award).

Stergios I. Roumeliotis received the Diploma in Electrical Engineering from the National Technical University of Athens, Greece, in 1995, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Southern California, CA in 1999 and 2000 respectively. From 2000 to 2002 he was a Postdoctoral Fellow at the California Institute of Technology, CA. Between 2002 and 2013 he was first an Assistant and then an Associate Professor with the Department of Computer Science and Engineering, University of Minnesota, MN, where he is currently a Professor. Since 2009, Dr. Roumeliotis is the Associate Director for Research of the Digital Technology Center (DTC). His research interests include distributed estimation under processing and communication constraints, active sensing for reconfigurable networks of sensors, and vision-aided inertial navigation for space, aerial, and ground vehicles, as well as mobile devices.

Dr. Roumeliotis is the recipient of the Guillermo E. Borja Award (2009), the National Science Foundation (NSF) Presidential Early Career Award for Scientists and Engineers (PECASE) (2008), the NSF CAREER award (2006), the McKnight Land-Grant Professorship award (2006-08), the ICRA Best Reviewer Award (2006), and he is the co-recipient of the One NASA Peer award (2006), and the One NASA Center Best award (2006). Papers he has co-authored have received the King-Sun Fu Best Paper Award of the IEEE Transactions on Robotics (2009), the Robotics Society of Japan Best Journal Paper award (2007), the ICASSP Best Student Paper award (2006), the NASA Tech Briefs award (2004), and three of them were Finalists for the RSS Best Paper Award (2009), the ICRA Best Student Paper Award (2009) and the IROS Best Paper Award (2006). Dr. Roumeliotis served as Associate Editor for the IEEE Transactions on Robotics between 2006 and 2010.