Information-Based Georeferencing by Dual State Iterated Extended Kalman Filter with Implicit Measurement Equations and Nonlinear Geometrical Constraints

Rozhin Moftizadeh, Johannes Bureick, Sören Vogel, Ingo Neumann and Hamza Alkhatib

June 9, 2020

Information-Based Georeferencing by Dual State Iterated Extended Kalman Filter with Implicit Measurement Equations and Nonlinear Geometrical Constraints

1st Rozhin Moftizadeh
Geodetic Institute
Leibniz Universität Hannover
Hanover, Germany
moftizadeh@gih.uni-hannover.de

2nd Johannes Bureick
Geodetic Institute
Leibniz Universität Hannover
Hanover, Germany
bureick@gih.uni-hannover.de

3rd Sören Vogel
Geodetic Institute
Leibniz Universität Hannover
Hanover, Germany
vogel@gih.uni-hannover.de

4th Ingo Neumann
Geodetic Institute
Leibniz Universität Hannover
Hanover, Germany
neumann@gih.uni-hannover.de

5th Hamza Alkhatib
Geodetic Institute
Leibniz Universität Hannover
Hanover, Germany
alkhatib@gih.uni-hannover.de

***Abstract*—Multi-Sensor-System (MSS) georeferencing is a challenging task in engineering that should be dealt with in the most accurate way possible. The easiest and most straightforward way for this purpose is to rely on Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU) data. However, at indoor environments or crowded inner-city areas, such data are not accurate to be entirely relied on. Therefore, appropriate filtering algorithms are required to compensate for possible errors and to improve the accuracy of the results. Sometimes it is also possible to increase the functionality of a filtering technique by engaging additional complementary information that can directly influence the outputs. Such information could be, e.g. geometrical features of the environment in which the MSS runs through. The current paper deals with MSS georeferencing by means of a Dual State Iterated Extended Kalman Filter (DSIEKF) that is based on an efficient combination of the Iterated Extended Kalman Filter (IEKF) with implicit measurement equations technique and nonlinear geometrical constraints. Final results of such an algorithm are shown to be satisfactory not only from the accuracy point of view but also the computation time.**

***Index Terms*—georeferencing, MSS, geometrical constraints, Iterated Extended Kalman Filter, Dual State, 6-DOF, Kalman filtering, Monte Carlo simulation**

I. Introduction

In engineering, Multi Sensor Systems (MSS) – which are various installed sensors on a single platform – are frequently used to capture different aspects of an environment. To combine the derived data for further analysis purposes, it is essential to know the position and orientation of the MSS with respect to a superordinate coordinate system. In other words, it is important for the MSS to be georeferenced. These position and orientation – that each consist of three parameters – are also referred to as the six Degrees of Freedom (6-DOF) or pose parameters. Also, the MSS velocities and accelerations at each epoch in time are of interest.

The easiest and typical way of georeferencing is to rely on Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU) data. In general, the accuracy of GNSS data lies within meter level; unless, differential techniques are applied that can increase the accuracy up to a centimeter or even millimeter level. However, in urban areas, due to shadowing and multipath effects, the accuracy is greatly affected, and a total signal loss is also possible. On the other hand, a low-cost IMU can obtain orientation results with an accuracy of $0.8°$ for heading angle and $0.1°$ for roll and pitch angles. Also, the IMU data are always in danger of drifting and therefore they should not be completely relied on [1]. Therefore, it is of great importance to develop an algorithm that can properly compensate for such errors.

One solution to the problem mentioned above is a Linear Kalman Filter (LKF), which could be applied on GNSS and IMU data, exclusively. However, due to a possible decreased number of such data in challenging environments, no significant improvement should be expected. Another solution is to increase the navigation performance by using additional sensors on the platform, including cameras and laser scanners, as well as map data. The performance, in this case, refers to not only accuracy but also integrity, continuity and availability, which is well discussed and processed in [2]. In [3], an Iterated Extended Kalman Filter (IEKF) technique with implicit measurement equations is proposed. The algorithm can deal well with GNSS and IMU errors by taking into consideration data from a highly accurate laser scanner along with all possible geometrical information of the environment.

This research was funded by the German Research Foundation (DFG) as part of the Research Training Group [RTG 2159].

The algorithm is then adapted to a similar case in [1] for georeferencing of an Unmanned Aerial Vehicle (UAV) and has further proven the method to be reliable. In the current paper, the proposed algorithm in [3] is modified to grant efficiency while preserving the accuracy of the results. This modified version is named Dual State Iterated Extended Kalman Filter (DSIEKF) in which the unknown parameters are divided into two different parts. A part that is fixed in size and comprises the pose parameters of the MSS and a part that includes the additional geometrical features, which can change in size from one epoch to the other. The second part usually depends on the extracted information from a reliable source such as a 3D city model. Therefore, its estimation is faster than the first part resulting in a faster yet accurate georeferencing algorithm.

The paper is organized as follows. In section II, a summary of the related researches to the current paper is given. In section III, an overview of LKF, IEKF, and DSIEKF is presented. Section IV is dedicated to the application of the mentioned algorithms on a simulated environment. In section V, a summary over the paper along with highlighted conclusions and an outlook over future work in the same area are given.

II. Related Work

There are several ways for MSS georeferencing that could be chosen based on the whole measurement scenario such as the type, number and accuracy of the installed sensors on the platform and the environment in which the MSS runs through. For outdoor applications, georeferencing could be done directly (sensor-driven), indirectly (target-driven), or by means of available referenced data sets (data-driven) [4] and [5]. For indoor applications, the classification is different, which is presented in detail by [6]. As the focus of the current paper relies on outdoor applications, related researches to the first classification are highlighted in the following.

"Direct (sensor-driven) georeferencing" could be done by using sensors such as GNSS receiver [5] and IMU [7] that directly deliver the 6-DOF with respect to a superordinate coordinate system. It is also possible to set up an arbitrary superordinate coordinate system by means of a total station or a laser tracker and to derive the pose of a certain MSS with respect to it [8] and [9]. In "indirect (target-driven) georeferencing", the MSS pose is derived with respect to targets that have already been referenced to a superordinate coordinate system. In this case, the targets could be flat markers with specific patterns [10] or simple 3D geometries such as cylinders or spheres [11]. Georeferencing could also be done by using available referenced data sets, which is also referred to as "data-driven georeferencing". In this type, the 6-DOF are matched with available georeferenced data sets such as 3D point clouds [12], digital surface models or 3D city models [6], [13], and [14].

On the other hand, filtering techniques are tools to compensate for possible errors within data sets derived from various sensors. One of such techniques, which is well-known and has been the basis of many pose estimation algorithms, is Kalman Filtering (KF). If both system and measurement equations are linear, the filter is referred to as LKF; however, sometimes it is also possible to have nonlinear system and measurement equations. In that case, another realization of KF called Extended Kalman Filter (EKF) is used. In this realization, the nonlinearity is overcome by means of Taylor series expansion around a certain state [15]. In research from [16], which is focused on sensor fusion of UAV's local sensors, the GNSS and IMU data are combined by means of EKF. Reference [17] has used EKF for collaboration of several UAVs in order to increase the accuracy of the estimated pose parameters by combining multiple Simultaneous Localization And Mapping (SLAM) algorithms. When linearizing around a certain state in EKF to deal with nonlinearity, it is possible to induce large errors in the filtering procedure, which can affect its convergence or efficiency. Therefore, a re-linearization around the updated state can be done to overcome the problem. Such a procedure is referred to as IEKF. So far researches have been mainly focused on IEKF with explicit measurement equations. In such equations, observations and states are separated from each other. This kind of equations are also referred to as Gauss-Markov-Models (GMM). However, sometimes the observations and the states are interconnected. These implicit measurement equations are so-called Gauss-Helmet-Models (GHM) that are used within IEKF by [18] and [19]. Reference [3] has used - for the first time - the IEKF method with implicit measurement equations for the purpose of MSS georeferencing. Reference [1] has applied the proposed method by [3] to georeference a UAV, which has further proven the algorithm to be reliable. Sometimes, it is possible to have changes in the system behaviour or its surrounding environment causing the system or observation model to vary over time. In that case, estimations are not limited to only the system pose but also the model(s) parameters that should be done simultaneously due to their dependency. One of the methods to deal with such a problem is called Dual State (DS) estimation, which is based on grouping the unknowns into two vectors using each to estimate the other, iteratively, as applied by [20], [21], [22], [23], and [24].

Moreover, sometimes the environment has additional information to offer, which could be integrated in the filtering process to increase the filtered results' accuracy. Such information is generally referred to as "constraints", and in case of geometrical information, it is called "geometrical constraints". Geometrical constraints of a scene could be the perpendicular facades of a building or the intersection line between facades of two adjacent buildings, etc. [3]. The constraints could be linear or nonlinear depending on the mathematical equations that describe them. In [25], a good overview of the possible algorithms in KF to consider linear and nonlinear constraints is presented. To deal with nonlinear constraints within filtering algorithms, it is generally suggested to linearize the mathematical equations and proceed to solve the problem by various algorithms that are already developed for linear cases (e.g. [26], [27], [28], and [29]). In [30], a maximum-likelihood-based filtering algorithm is suggested that can deal with both linear and nonlinear constraints without the necessity for linearization in case of nonlinearity. In research from [31], linear and nonlinear systems subject to linear and nonlinear constraints are studied, and different algorithms are developed that could be used in any of such cases. In [32], an algorithm is proposed that can deal with nonlinear equality constraints; however, the method can only handle second-order scalar constraints. Reference [33] has suggested a smoothly constrained KF that could be used for nonlinear constraints of any type. In [3] and [34], nonlinear equality and inequality geometrical constraints are integrated into the IEKF with implicit measurement equations for MSS georeferencing by means of projection and Power Density Function (PDF) truncation methods, respectively. Sometimes it is probable to have a large number of geometrical constraints in an environment that can cause the proposed algorithm in [3] inefficient from the computation time aspect. Therefore, the current paper has focused on adapting this algorithm to DS estimation method that is applied to a simulated environment for georeferencing a UAV. Furthermore, both LKF and the algorithm in [3] have also been applied for better judgment of the functionality of this new algorithm that is named DSIEKF.

III. Methodology

Current paper deals with georeferencing a UAV, which is equipped with a GNSS receiver, an IMU, and a 3D laser scanner. It is assumed that the system moves through an urban area with multiple building models, which are constantly captured by the laser scanner. The whole MSS and its surrounding environment are simulated, which is further explained in IV-A.

A. Linear Kalman Filter

LKF is the most straightforward realization of KF that is used in case of GMM with linear system and observation models as follows:

$$\begin{aligned} \mathbf{l}_k + \boldsymbol{\nu}_k &= \mathbf{H}_{x,k}\, \mathbf{x}_k \\ \mathbf{x}_k &= \mathbf{F}_{x,k-1}\, \mathbf{x}_{k-1} + \mathbf{F}_{u,k-1}\, \mathbf{u}_{k-1} + \boldsymbol{\omega}_{k-1} \end{aligned} \tag{1}$$

wherein k is the epoch and $\mathbf{l}$, $\mathbf{x}$, $\mathbf{H}_x$, $\mathbf{F}_x$, $\mathbf{F}_u$, $\mathbf{u}$, $\boldsymbol{\omega}$, $\boldsymbol{\nu}$ are the measurement vector, state vector, design matrix, transition matrix, control matrix, control vector, system noise, and measurement noise, respectively.

In the current paper, the observation model of LKF is as follows:

$$\mathbf{l}_k + \boldsymbol{\nu}_k = \mathbf{x}_k \tag{2}$$

The observation and state vectors in the current paper are defined according to (3).

$$\mathbf{l}_k = \left[\mathbf{l}_k^G; \mathbf{l}_k^I\right],\ \mathbf{x}_k = \left[\mathbf{x}_k^P; \mathbf{x}_k^O; \mathbf{x}_k^V\right] \tag{3}$$

wherein $\mathbf{l}^G$ and $\mathbf{l}^I$ are vectors consisting of the 3 positions (x, y, z) and 3 orientations (ω, φ, κ), which are derived from GNSS receiver and IMU, respectively. It should be noted that Euler angles suffer from discontinuity and singularity problems and therefore they are not stable. On the other hand, quaternions can better handle such problems. However, for better interpretation of the simulation results in the current paper, the rotation representation by means of quaternions is avoided and instead, rotation matrices are used within the algorithms to deal with the aforementioned challenges of Euler angles. $\mathbf{x}^P$, $\mathbf{x}^O$, and $\mathbf{x}^V$ are parameter vectors containing three translations (t_x, t_y, t_z), three orientations (ω, φ, κ), and three velocities (V_x, V_y, V_z), respectively. In this case, the number of observations is less than the number of unknowns. In other words, for the velocity parameters, no measurements exist. Therefore, those columns within the design matrix ($\mathbf{H}_x$) that correspond to these parameters are filled with zeros.

For behaviour of the considered MSS in this paper, a simple linear model is assumed with a transition matrix according to (4).

$$\mathbf{F}_{x,state} = \begin{bmatrix} \mathbf{I}_{[3\times3]} & \mathbf{0}_{[3\times3]} & diag([\Delta\tau, \Delta\tau, \Delta\tau]) \\ \mathbf{0}_{[6\times3]} & & \mathbf{I}_{[6\times6]} \end{bmatrix} \tag{4}$$

$\mathbf{I}$ is an identity matrix and $\Delta\tau$ is the time period between two consecutive epochs, which is set to 1 second in the context of this paper.

B. Iterated Extended Kalman Filter with Nonlinear Equality Constraints

Sometimes due to nonlinear behaviour of the system or environment, it is possible to have nonlinear equations, which are usually linearized for further processing. Such a linearization – that is done by Taylor series expansion around a certain state – can induce large errors in the filtering process, which leads to divergence and inefficiency problems. Therefore, a re-linearization around the updated states could be done to avoid such consequences. This realization of EKF is usually referred to as IEKF. The filtering procedure gets even more complicated when the observations and unknowns are interconnected, which means that they cannot be separated from each other. Having such a GHM within IEKF is the main scope of [3]. The observation and system models, in this case, have a general form as given in (5).

$$\begin{aligned} \mathbf{h}(\mathbf{l}_k + \boldsymbol{\nu}_k, \mathbf{x}_k) &= \mathbf{0} \\ \mathbf{x}_k &= \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \boldsymbol{\omega}_{k-1}) \end{aligned} \tag{5}$$

wherein, $\mathbf{h}$, $\mathbf{f}$ are the measurement condition equation and nonlinear system model, respectively.

As previously mentioned, in this paper, it is assumed that the MSS platform contains not only the GNSS and IMU sensors but also a 3D laser scanner, which constantly captures the surrounding environment. On the other hand, corner points of the simulated building models are considered as additional observations to ensure higher redundancy and thus more accurate estimations. Therefore, the observation model of IEKF consists of two types. The first type that is used for GNSS and IMU observations and is the same as defined in III-A. The second type that is used for 3D scanned data from the laser scanner and corner points of the 3D city model, which is the equation of a plane and could be written as follows:

$$\mathbf{n}_{\mathbf{x}k}.\mathbf{X}_k + \mathbf{n}_{\mathbf{y}k}.\mathbf{Y}_k + \mathbf{n}_{\mathbf{z}k}.\mathbf{Z}_k - \mathbf{d}_k = \mathbf{0} \tag{6}$$

wherein $\mathbf{n_x}$, $\mathbf{n_y}$, and $\mathbf{n_z}$ are the planes' normal vector parameters. $\mathbf{d}$ is the planes' distance parameter vector. $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ are the transformed laser scanner 3D point cloud from local to global coordinate system. The transformation is done according to (7). In case of corner points, no transformation is required.

$$\mathbf{P}_{glo,k} = \mathbf{t}_k + \mathbf{R}_k \mathbf{P}_{loc,k} \tag{7}$$

wherein, $\mathbf{P}_{glo}$ is the transformed laser scanner point cloud from local to global coordinate system. $\mathbf{t}$ and $\mathbf{R}$ are the translation parameters vector and rotation matrix, respectively. $\mathbf{P}_{loc}$ is the laser scanner point cloud in its local coordinate system.

The observation and state vectors are extended according to (8).

$$\mathbf{l}_k = \left[\mathbf{l}_k^{LS}; \mathbf{l}_k^{CP}; \mathbf{l}_k^{G}; \mathbf{l}_k^{I}\right], \ \mathbf{x}_k = \left[\mathbf{x}_k^{P}; \mathbf{x}_k^{O}; \mathbf{x}_k^{V}; \mathbf{x}_k^{PL}\right] \tag{8}$$

wherein $\mathbf{l}^{LS}$, $\mathbf{l}^{CP}$, and $\mathbf{x}^{PL}$ are the local laser scanner observations, the corner points coordinates, and the planes' parameters, respectively.

Transition matrix, in this case, is as follows:

$$\mathbf{F}_x = \begin{bmatrix} \mathbf{F}_{x,state_{[9\times 9]}} & \mathbf{0}_{[9\times 4\cdot E_k]} \\ \mathbf{0}_{[4\cdot E_k \times 9]} & \mathbf{I}_{[4\cdot E_k \times 4\cdot E_k]} \end{bmatrix} \tag{9}$$

wherein, $\mathbf{F}_{x,state}$ is defined as (4) and E_k is the total number of planes that are detected in epoch k.

Nonlinear geometrical constraints that are considered for the current paper are the unity of building planes' normal vectors within the simulated environment [1]. Such a constraint is mathematically defined as (10).

$$\sqrt{n_x^2 + n_y^2 + n_z^2} = 1 \tag{10}$$

wherein, n_x, n_y, and n_z are the normal vector parameters of each plane.

The whole IEKF formulation and algorithm, as well as the method for applying the geometrical constraints, is taken from [3].

C. Dual State Iterated Extended Kalman Filter with Nonlinear Equality Constraints

Sometimes the system behaviour might change over time, causing the system model to differ from one epoch to the other. Also, it is possible to have changing observation model over time due to environmental conditions. Therefore, it might be necessary to estimate not only the system state but also the engaging model(s) parameters, simultaneously. Since these two estimations are codependent, one solution is to apply dual estimation methods. In such methods, the estimation of these two groups of parameters is done by alternating between using the model to estimate the system state and using the system state to estimate the model [21].

As it was mentioned in III-B, one type of observation models in this paper is the plane equation that should be satisfied in each epoch after estimating the system states. On the other hand, according to (10), the considered geometrical constraints require plane parameters modification. Therefore, the system states and observation model parameters are interconnected and subject to change over time. Such codependency is accounted for within the IEKF algorithm given in [3] by including the system states and planes parameters all in one vector, as shown in (8). However, the number of captured planes by the laser scanner might increase over time, causing the state vector to get larger, which in turn leads to longer computation time. Therefore, in the current paper, instead of putting all the unknown parameters in one vector, they are divided in two vectors shown in (11) and (12) along with their related observations.

$$\mathbf{l}_k^{(1)} = \left[\mathbf{l}_k^{LS}; \mathbf{l}_k^{G}; \mathbf{l}_k^{I}\right], \ \mathbf{x}_k^{(1)} = \left[\mathbf{x}_k^{P}; \mathbf{x}_k^{O}; \mathbf{x}_k^{V}\right] \tag{11}$$

$$\mathbf{l}_k^{(2)} = \left[\mathbf{l}_k^{LS}; \mathbf{l}_k^{CP}\right], \ \mathbf{x}_k^{(2)} = \left[\mathbf{x}_k^{PL}\right] \tag{12}$$

wherein, $\mathbf{x}^{(1)}$ and $\mathbf{l}^{(1)}$, $\mathbf{x}^{(2)}$ and $\mathbf{l}^{(2)}$ are the system states vector and its related observations, and parameters vector of the observation model and its related measurements, respectively. $\mathbf{x}^{(1)}$ is fixed in size and contains pose parameters of the MSS whereas $\mathbf{x}^{(2)}$ contains the captured planes' parameters and can change in size from one epoch to the other. Since estimation of $\mathbf{x}^{(2)}$ depends on reliable information from the environment, its convergence is faster than $\mathbf{x}^{(1)}$, which results in less overall computation time. This new algorithm is named DSIEKF and its pseudo-code is given in Algorithm 1. Main idea of this algorithm is similar to that in [3] except that the explained procedure in III-B is applied twice. Once for updating the system states vector (lines 23 to 30) and once for updating the planes parameters (lines 32 to 42). In each iteration, the estimated system states are used to filter the planes parameters that are then used to update the system states in the next iteration. Unlike the algorithm in [3] where the "constraint step" is applied on the final filtered results, in this case, it is applied on the estimated planes parameters in each iteration (lines 37 to 41). Finally, in each epoch, indices of the captured planes along with their filtered parameters and VCM are stored in a matrix (line 49) that is further investigated in the next epoch. In each epoch, if indices of the captured planes already exist in this matrix, their parameters are treated as deterministic values; otherwise, they are filtered (line 10). Consequently, each plane's parameters are filtered only once and then used as deterministic values in the next epochs. Moreover, the generated GNSS and IMU data of the first epoch along with their corresponding VCM are used for initializing $\mathbf{x}^{(1)}$ and its VCM. Figure 1 shows the flow diagram of the algorithm.

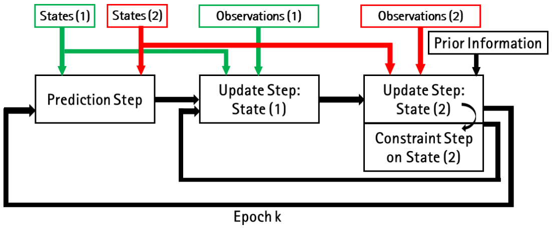

Fig. 1. Flow diagram of the DSIEKF algorithm.

Algorithm 1: Dual State Iterated Extended Kalman Filter (DSIEKF) with nonlinear equality constraints.

```
1  System model 1:
     $\mathbf{x}_k^{(1)} = \mathbf{f}^{(1)}(\mathbf{x}_{k-1}^{(1)}, \mathbf{u}_{k-1}^{(1)}, \boldsymbol{\omega}_{k-1}^{(1)}), \quad \boldsymbol{\omega}_{k-1}^{(1)} \sim N\left(\mathbf{0}, \boldsymbol{\Sigma}_{\omega\omega}^{(1)}\right)$
2  System model 2:
     $\mathbf{x}_k^{(2)} = \mathbf{f}^{(2)}(\tilde{\mathbf{x}}_k^{(2)}, \mathbf{u}_k^{(2)}, \boldsymbol{\omega}_k^{(2)}), \quad \boldsymbol{\omega}_k^{(2)} \sim N\left(\mathbf{0}, \boldsymbol{\Sigma}_{\omega\omega}^{(2)}\right)$
3  Observation model 1:
     $\mathbf{h}^{(1)}(\mathbf{l}_k^{(1)} + \boldsymbol{\nu}_k^{(1)}, \mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}) = 0, \quad \boldsymbol{\nu}_k^{(1)} \sim N\left(\mathbf{0}, \boldsymbol{\Sigma}_{\nu\nu}^{(1)}\right)$
4  Observation model 2:
     $\mathbf{h}^{(2)}(\mathbf{l}_k^{(2)} + \boldsymbol{\nu}_k^{(2)}, \mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}) = 0, \quad \boldsymbol{\nu}_k^{(2)} \sim N\left(\mathbf{0}, \boldsymbol{\Sigma}_{\nu\nu}^{(2)}\right)$
5  Initialization state vector 1: $\hat{\mathbf{x}}_0^{(1+)} = \mathbf{x}_0^{(1)}, \ \boldsymbol{\Sigma}_{\hat{x}\hat{x},0}^{(1+)} = \boldsymbol{\Sigma}_{xx,0}^{(1)}, \quad k = 1$
6  Initialization of a matrix for saving the filtered second state vector:
     $\mathbf{C} = [\,]$
7  while $k < \mathrm{K}$ do
8      Extraction of Captured Planes Data from the 3D City Model
9      $\hat{\mathbf{x}}_0^{(2+)} = \mathbf{x}_0^{(2)}, \ \boldsymbol{\Sigma}_{\hat{x}\hat{x},0}^{(2+)} = \boldsymbol{\Sigma}_{xx,0}^{(2)}$
10     Only those planes that are not in $\mathbf{C}$ will be filtered. Otherwise,
         they are treated as deterministic values. In other words, each
         plane's parameters are filtered only once. The initial values and
         VCM are taken from available information.
11     Prediction Step
12     $\mathbf{F}_{x,k}^{(1)} = \partial \mathbf{F}^{(1)} / \partial \mathbf{x}^{(1)}|_{\hat{\mathbf{x}}_{k-1}^{(1+)}, \mathbf{u}_{k-1}^{(1)}, \boldsymbol{\omega}_{k-1}^{(1)}}$
13     $\mathbf{F}_{\omega,k}^{(1)} = \partial \mathbf{F}^{(1)} / \partial \omega^{(1)}|_{\hat{\mathbf{x}}_{k-1}^{(1+)}, \mathbf{u}_{k-1}^{(1)}, \boldsymbol{\omega}_{k-1}^{(1)}}$
14     $\hat{\mathbf{x}}_k^{(1-)} = \mathbf{F}_{x,k}^{(1)} \hat{\mathbf{x}}_{k-1}^{(1+)} + \boldsymbol{\omega}_{k-1}^{(1)}$
15     $\boldsymbol{\Sigma}_{xx,k}^{(1-)} = \mathbf{F}_{x,k}^{(1)} \boldsymbol{\Sigma}_{\hat{x}\hat{x},k-1}^{(1+)} \mathbf{F}_{x,k}^{(1)^T} + \mathbf{F}_{\omega,k}^{(1)} \boldsymbol{\Sigma}_{\omega\omega}^{(1)} \mathbf{F}_{\omega,k}^{(1)^T}$
16     $\mathbf{f}_{x,k}^{(2)} = \partial \mathbf{f}^{(2)} / \partial \mathbf{x}^{(2)}|_{\hat{\mathbf{x}}_k^{(2+)}, \mathbf{u}_k^{(2)}, \boldsymbol{\omega}_k^{(2)}}$
17     $\mathbf{f}_{\omega,k}^{(2)} = \partial \mathbf{f}^{(2)} / \partial \omega^{(2)}|_{\hat{\mathbf{x}}_k^{(2+)}, \mathbf{u}_k^{(2)}, \boldsymbol{\omega}_k^{(2)}}$
18     $\hat{\mathbf{x}}_k^{(2-)} = \mathbf{f}_{x,k}^{(2)} \hat{\mathbf{x}}_k^{(2+)} + \boldsymbol{\omega}_k^{(2)}$
19     $\boldsymbol{\Sigma}_{xx,k}^{(2-)} = \lambda \boldsymbol{\Sigma}_{\hat{x}\hat{x},k}^{(2+)}, \quad \lambda \in (0, 1]$
20     Update Step
21     $\breve{\mathbf{l}}_{k,0}^{(1)} = \mathbf{l}_k^{(1)}, \ \breve{\mathbf{x}}_{k,0}^{(1)} = \hat{\mathbf{x}}_k^{(1-)}, \ \breve{\mathbf{l}}_{k,0}^{(2)} = \mathbf{l}_k^{(2)}, \ \breve{\mathbf{x}}_{k,0}^{(2)} =$
         $\hat{\mathbf{x}}_k^{(2-)}, \ \boldsymbol{\Delta}\mathbf{x} = \infty, \ n = 0$
22     for $m = 0 \ldots M - 1$ do
23         $\mathbf{h}_{x,k,m}^{(1)} = \partial \mathbf{h}^{(1)} / \partial \mathbf{x}^{(1)}|_{\breve{\mathbf{l}}_{k,m}^{(1)}, \breve{\mathbf{x}}_{k,m}^{(1)}, \breve{\mathbf{x}}_{k,m}^{(2)}}$
24         $\mathbf{h}_{l,k,m}^{(1)} = \partial \mathbf{h}^{(1)} / \partial \mathbf{l}^{(1)}|_{\breve{\mathbf{l}}_{k,m}^{(1)}, \breve{\mathbf{x}}_{k,m}^{(1)}, \breve{\mathbf{x}}_{k,m}^{(2)}}$
25         $\mathbf{O}_{k,m}^{(1)} = \mathbf{h}_{x,k,m}^{(1)} \boldsymbol{\Sigma}_{xx,k}^{(1-)} \mathbf{H}_{x,k,m}^{(1)^T}$
26         $\mathbf{S}_{k,m}^{(1)} = \mathbf{h}_{l,k,m}^{(1)} \boldsymbol{\Sigma}_{\nu\nu}^{(1)} \mathbf{H}_{l,k,m}^{(1)^T}$
27         $\mathbf{K}_{k,m}^{(1)} = \boldsymbol{\Sigma}_{xx,k}^{(1-)} \mathbf{H}_{x,k,m}^{(1)^T} (\mathbf{O}_{k,m}^{(1)} + \mathbf{S}_{k,m}^{(1)})^{-1}$
28         $\mathbf{r}_{k,m}^{(1)} = \mathbf{h}_{l,k,m}^{(1)} (\mathbf{l}_k^{(1)} - \breve{\mathbf{l}}_{k,m}^{(1)}) + \mathbf{h}_{x,k,m}^{(1)} (\hat{\mathbf{x}}_k^{(1-)} - \breve{\mathbf{x}}_{k,m}^{(1)})$
             $\breve{\mathbf{x}}_{k,m+1}^{(1)} = \hat{\mathbf{x}}_k^{(\mathbf{1}-)} - \mathbf{K}_{k,m}^{(1)} (\mathbf{h}^{(1)} (\breve{\mathbf{l}}_{k,m}^{(1)}, \breve{\mathbf{x}}_{k,m}^{(1)}, \breve{\mathbf{x}}_{k,m}^{(2)}) + \mathbf{r}_{k,m}^{(1)})$
29         $\mathbf{G}_{k,m}^{(1)} = \boldsymbol{\Sigma}_{\nu\nu}^{(1)} \mathbf{H}_{l,k,m}^{(1)^T} (\mathbf{O}_{k,m}^{(1)} + \mathbf{S}_{k,m}^{(1)})^{-1}$
30         $\breve{\mathbf{l}}_{k,m+1}^{(1)} = \mathbf{l}_k^{(1)} - \mathbf{G}_{k,m}^{(1)} (\mathbf{h}^{(1)} (\breve{\mathbf{l}}_{k,m}^{(1)}, \breve{\mathbf{x}}_{k,m}^{(1)}, \breve{\mathbf{x}}_{k,m}^{(2)}) + \mathbf{r}_{k,m}^{(1)})$
31         while $\max(\boldsymbol{\Delta}\mathbf{x}) > c_{stop}$ do
32             $\breve{\mathbf{l}}_{k,n}^{(2)} = \breve{\mathbf{l}}_{k,m}^{(2)}, \ \breve{\mathbf{x}}_{k,n}^{(2)} = \breve{\mathbf{x}}_{k,m}^{(2)}$
33             Repeat lines 23 to 30 for the second states vector.
34             $\mathbf{L}_{k,n}^{(2)} = \mathbf{I}_{u_2 \times u_2} - \mathbf{K}_{k,n}^{(2)} \mathbf{h}_{x,k,n}^{(2)}$
35             $\boldsymbol{\Sigma}_{\hat{x}\hat{x},k,n+1}^{(2+)} = \mathbf{L}_{k,n}^{(2)} \boldsymbol{\Sigma}_{xx,k}^{(2-)} \mathbf{L}_{k,n}^{(2)^T} + \mathbf{K}_{k,n}^{(2)} \mathbf{S}_{k,n}^{(2)} \mathbf{K}_{k,n}^{(2)^T}$
36             Constraint Step
37             $\mathbf{D} = g'(\hat{\mathbf{x}}_k^{(2-)})$
38             $\mathbf{d} = \mathbf{b} - g(\hat{\mathbf{x}}_k^{(2-)}) + g'(\hat{\mathbf{x}}_k^{(2-)}) \hat{\mathbf{x}}_k^{(2-)}$
39             $\mathbf{W} = \underset{j \times j}{\mathbf{I}}$
40             $\breve{\mathbf{x}}_{k,n+1}^{(2)} =$
                 $\breve{\mathbf{x}}_{k,n+1}^{(2)} - \mathbf{W}^{-1} \mathbf{D}^T (\mathbf{D} \mathbf{W}^{-1} \mathbf{D}^T)^{-1} (\mathbf{D} \breve{\mathbf{x}}_{k,n+1}^{(2)} - \mathbf{d})$
41             $\boldsymbol{\Sigma}_{\hat{x}\hat{x},k,n+1}^{(2+)} = \boldsymbol{\Sigma}_{\hat{x}\hat{x},k,n+1}^{(2+)} -$
                 $\boldsymbol{\Sigma}_{\hat{x}\hat{x},k,n+1}^{(2+)} \mathbf{D}^T (\mathbf{D} \boldsymbol{\Sigma}_{\hat{x}\hat{x},k,n+1}^{(2+)} \mathbf{D}^T)^{-1} \mathbf{D} \boldsymbol{\Sigma}_{\hat{x}\hat{x},k,n+1}^{(2+)}$
42             $n = n + 1$
43         $\breve{\mathbf{x}}_{k,m+1}^{(2)} = \breve{\mathbf{x}}_{k,n+1}^{(2)}, \ \breve{\mathbf{l}}_{k,m+1}^{(2)} = \breve{\mathbf{l}}_{k,n+1}^{(2)}, \ \boldsymbol{\Delta}\mathbf{x} = \breve{\mathbf{x}}_{k,n+1}^{(2)} - \breve{\mathbf{x}}_{k,n}^{(2)}$
44     $\hat{\mathbf{x}}_k^{(1+)} = \breve{\mathbf{x}}_{k,M-1}^{(1)}, \ \hat{\mathbf{l}}_k^{(1+)} = \breve{\mathbf{l}}_{k,M-1}^{(1)}$
45     $\mathbf{S}_{k,M-1}^{(1)} = \mathbf{h}_{l,k,M-1}^{(1)} \boldsymbol{\Sigma}_{\nu\nu}^{(1)} \mathbf{H}_{l,k,M-1}^{(1)^T}$
         $\mathbf{l}_k^{(1)} = \mathbf{I}_{u_1 \times u_1}^{(1)} - \mathbf{K}_{k,M-1}^{(1)} \mathbf{h}_{x,k,M-1}^{(1)}$
46     $\boldsymbol{\Sigma}_{\hat{x}\hat{x},k}^{(1+)} = \mathbf{l}_k^{(1)} \boldsymbol{\Sigma}_{xx,k}^{(1-)} \mathbf{L}_k^{(1)^T} + \mathbf{K}_{k,M-1}^{(1)} \mathbf{S}_{k,M-1}^{(1)} \mathbf{K}_{k,M-1}^{(1)^T}$
47     $\hat{\mathbf{x}}_k^{(2+)} = \breve{\mathbf{x}}_{k,n+1}^{(2)}, \ \hat{\mathbf{l}}_k^{(2+)} = \breve{\mathbf{l}}_{k,n+1}^{(2)}$
48     $\boldsymbol{\Sigma}_{\hat{x}\hat{x},k}^{(2+)} = \boldsymbol{\Sigma}_{\hat{x}\hat{x},k,n+1}^{(2+)}$
49     $\mathbf{C} = [\mathbf{C}; (\mathrm{idx}, \ \hat{\mathbf{x}}_k^{(2+)}, \ \boldsymbol{\Sigma}_{\hat{x}\hat{x},k}^{(2+)})]$, "idx" are indices of those planes
         that are captured in the current epoch.
```

IV. Application

A. Simulated Environment

In this paper, the simulated environment is taken from [1] and includes generated data for a 3D laser scanner, a GNSS receiver and an IMU – which are all assumed to be mounted on a UAV – along with unreal building models and a ground plane, which altogether represent a 3D city model. 3D city models are 3D digital models of cities that contain spatial and georeferenced data for different parts of a city (buildings, sites, etc.) [35]. These models exist in different levels of detail, which could be used for different purposes. In this paper a second Level of Detail (LoD2) 3D city model is resembled by the simulated building models and the ground plane. The main reason for that is the availability of such LoD2 models for all the cities in Germany when dealing with real case scenarios. In total, 70 epochs are assumed. The true positions depict a length of 70 meters in y-axis that is covered by the MSS in 70 epochs while ascending for 4 meters in z-axis. The GNSS and IMU observations are generated by adding normally distributed noise to the true positions and orientations. Aside from 3D scanned points by the laser scanner, indices of the building planes, to which they are assigned, are also known. Figure 2 shows an overview of the simulated environment. The dotted black lines in this figure represent scan lines of the UAV that moves through the whole environment in y direction. Table I gives the accuracy and system noise values which are considered in the simulated environment.

LKF, IEKF, and DSIEKF algorithms are applied on the simulated environment in 500 Monte Carlo (MC) runs. In each run, new GNSS and IMU observations are generated by adding various random noise to the true states. Such a task is required to ensure functionality and performance of the proposed algorithm subject to different observations.

TABLE I
ASSUMED ACCURACY AND SYSTEM NOISE VALUES WITHIN THE SIMULATED ENVIRONMENT.

Initial state accuracy	$\sigma_{P,0} =$	$0.5\,\mathrm{m}$
	$\sigma_{O,0} =$	$0.2°$
	$\sigma_{V,0} =$	$1\,\mathrm{m/s}$
	$\sigma_{n,0} =$	0.0001
	$\sigma_{d,0} =$	$0.001\,\mathrm{m}$
	$\sigma_{CP,0} =$	$0.0001\,\mathrm{m}$
System noise	$\Delta\tau =$	1 s
	$\sigma_{P,w} =$	0
	$\sigma_{O,\omega} =$	0
	$\sigma_{V,\omega} =$	$5 \cdot \Delta\tau$ m/s
	$\sigma_{n,\omega} =$	0
	$\sigma_{d,\omega} =$	0
	$\sigma_{CP,\omega} =$	0
Measurement noise	$\sigma_{LS} =$	$0.02\,\mathrm{m}$
	$\sigma_{P} =$	$0.5\,\mathrm{m}$
	$\sigma_{O} =$	$0.2°$
	$\sigma_{CP} =$	$0.0001\,\mathrm{m}$

B. Results

In this paper, two different scenarios are considered. In the first one, no subsampling is applied on the laser scanner

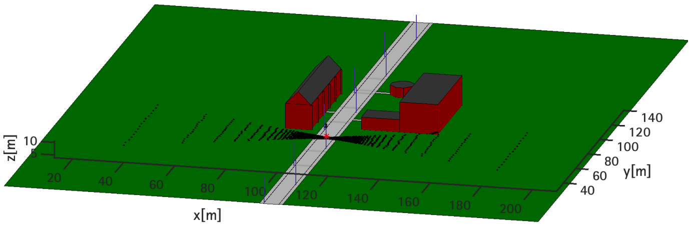

Fig. 2. 3D view of the simulated environment.

measurements neither in IEKF nor DSIEKF. The second scenario depicts results of DSIEKF when no subampling is applied versus the case where only 20% of the laser scanner measurements are considered in each epoch. These 20% observations are randomly selected from the whole. These scenarios are referred to as "scenario one" and "scenario two", respectively. Also, in the second scenario, "DSIEKF 1" refers to the case where no subsampling is applied whereas "DSIEKF 2" is used for the case with subsampling.

Figures 3 and 4 show the average Root Mean Square Error (RMSE) of the estimated translation parameters in each epoch over the MC runs for both scenarios. In each MC run, RMSE value is calculated as follows:

$$\mathbf{RMSE}_k = \sqrt{\frac{1}{N}\sum_{k=1}^{k=N}(\mathbf{x}_k^P - \bar{\mathbf{x}}_k^P)^2} \tag{13}$$

wherein $\mathbf{x}_k^P$ and $\bar{\mathbf{x}}_k^P$ are the estimated translation parameters vector in each epoch and their true values, respectively. N is the number of epochs up to the current one. Average RMSE over the MC runs is calculated according to (14).

$$\overline{\mathbf{RMSE}_k} = \frac{1}{S}\sum \mathbf{RMSE}_k \tag{14}$$

wherein S is the total number of MC runs, which is 500 in this paper.

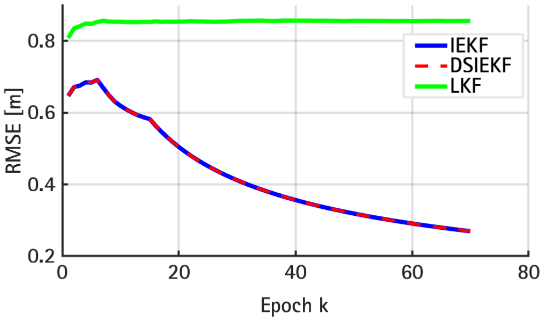

Fig. 3. Mean of RMSE for all translation parameters in all epochs over the total MC runs in scenario one.

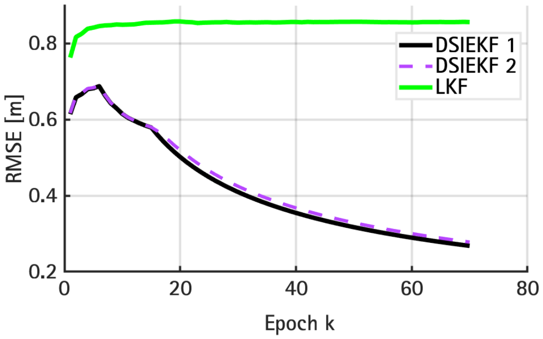

Fig. 4. Mean of RMSE for all translation parameters in all epochs over the total MC runs in scenario two.

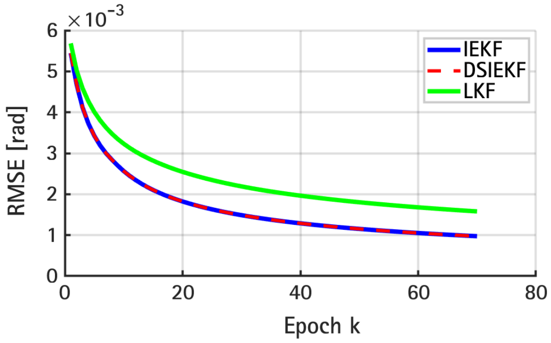

Fig. 5. Mean of RMSE for all orientation parameters in all epochs over the total MC runs in scenario one.

In both scenarios, it could be seen that LKF estimations are not satisfactory, which is due to considerably less number of observations compared to the other two algorithms. The reason for not having a decreasing pattern, in this case, is due to the selection of high velocity system noise compared to the observation noise in the system model. On the other hand, IEKF and DSIEKF have converged to the true values over time. Since in both of these algorithms, the laser scanner measurements and corner points data are taken as additional observations, and the high velocity system noise is also well dealt with. However, it could be seen that subsampling in scenario two has led to slightly larger RMSE values compared to the case where no subsampling is applied. The same pattern could also be seen for average RMSE measure of the orientation parameters as shown in figures 5 and 6. It could be seen that in case of orientations, due to the absence of angular velocity in the system model and therefore no corresponding

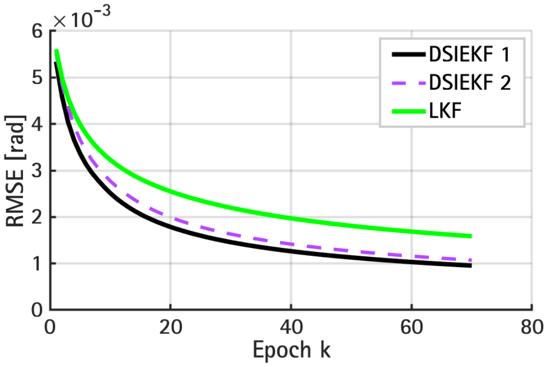

Fig. 6. Mean of RMSE for all orientation parameters in all epochs over the total MC runs in scenario two.

system noise, LKF has also a decreasing pattern.

Figure 7 visualizes the RMSE statistics of the translation parameters in scenario two over all the MC runs by means of box plots for DSIEKF cases. In each box plot, the middle line shows the median. Tops and bottoms are the 25th and 75th percentiles, respectively. Distances between these tops and bottoms are the interquartile ranges. The lines that are extended above and below each box plot are the whiskers. These whiskers are derived from the ends of the interquartile ranges to the furthest sample data within the whisker length. The red crosses are the outliers in the sense that their value is more than 1.5 times the interquartile range away from the top or bottom of the box. It could be seen that in both DSIEKF cases, the median, as well as the interquartile ranges, are large and they decrease over time. The reason for such behaviour is the small number of observations in the first few epochs. Over time, the number of measurements gets larger, which leads in turn to more accurate estimations and thus smaller RMSE values. Moreover, the median in each box plot is fairly located in the middle, which can represent a normal distribution with no skewness and thus having no systematic errors within the estimations as expected. Finally, the same behaviour with similar statistical estimations could be seen for both DSIEKF cases. Therefore, it could be concluded that although subsampling leads to slightly different mean and median values for RMSE, the overall statistical pattern remains unchanged. The same pattern could also be seen for the RMSE statistics of the orientation parameters as shown in figure 8.

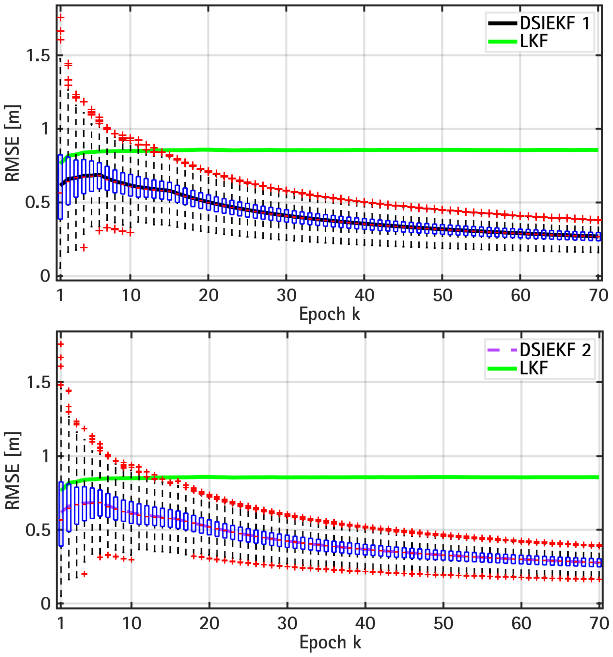

Fig. 7. Box plot of the mean of RMSE for all translation parameters in all epochs over the total MC runs in scenario two. Top: DSIEKF without subsampling. Bottom: DSIEKF with subsampling.

The maximum difference between the estimated translation and orientation parameters by IEKF and DSIEKF in scenario one are 7×10^{-5} meters and 9×10^{-7} radians, respectively. These estimation differences by IEKF and LKF are 0.68 meters and 0.0013 radians. In scenario two, the maximum difference between the estimated translation and orientation parameters by DSIEKF 1 and DSIEKF 2 are 0.26 meters and 7×10^{-4} radians, respectively. These estimation differences by DSIEKF 1 and LKF are 0.69 meters and 0.0014 radians.

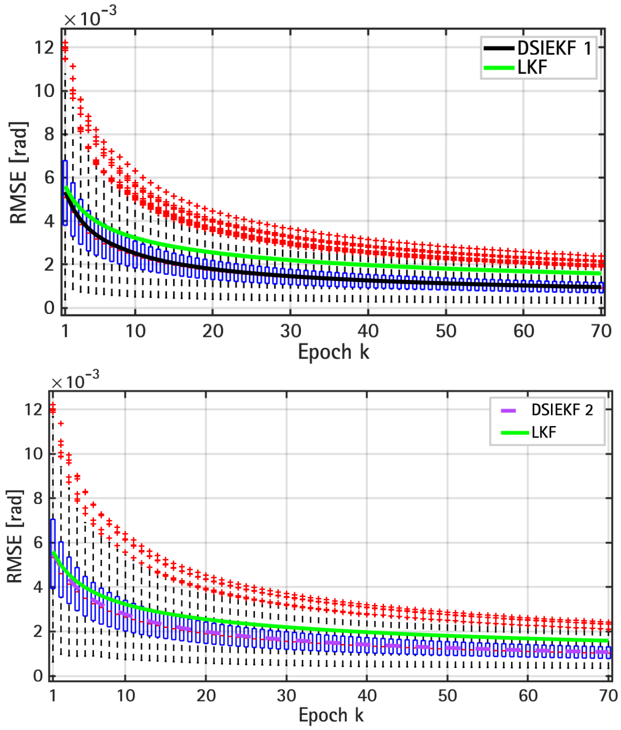

Fig. 8. Box plot of the mean of RMSE for all orientation parameters in all epochs over the total MC runs in scenario two. Top: DSIEKF without subsampling. Bottom: DSIEKF with subsampling.

Figures 9 and 10 show maximum absolute differences between the estimated planes parameters (normal vector and distance to origin) and their true values from the building models for both scenarios. It could be seen that DSIEKF delivers estimations that are closer to the true values than those derived by IEKF. In the second scenario, subsampling the laser scanner measurements has led to better estimations compared to the case where no subsampling is applied. Therefore, it is probable that measurement errors are propagated to the estimated parameters of the planes given that the laser scanner measurements are assumed to be free of outliers with correct assignments to the planes in the simulated building models. Also, it could be seen that in DSIEKF the parameters of the planes are estimated only in some epochs and remain constant for some time whereas in IEKF they are derived in every single epoch.

Figures 11 and 12 show the duration of update step in each epoch for both scenarios. In scenario one, it could be seen that in those epochs where planes parameters are estimated by DSIEKF, the update time is close to IEKF; however, in the other epochs where no such estimations are required, the

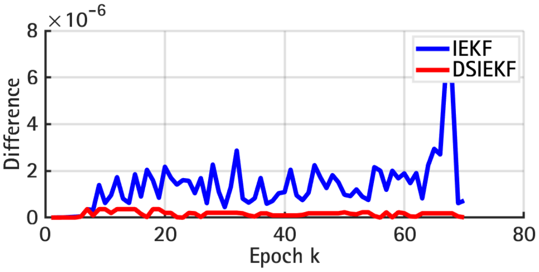

Fig. 9. Maximum absolute difference between the estimated planes parameters and their true values in scenario one.

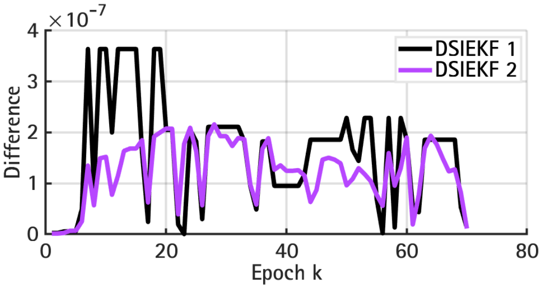

Fig. 10. Maximum absolute difference between the estimated planes parameters and their true values in scenario two.

update step is shorter. In scenario two, it could be seen that subsampling leads to a significantly shorter update time at the cost of losing accuracy in state estimations compared to the case with full laser scanner data. LKF in both scenarios has the shortest duration, which is expected due to considerably less amount of observations compared to the other cases. Total update time over 70 epochs in scenario one is approximately 13 minutes, 11 minutes, 0.1 seconds for IEKF, DSIEKF, and LKF, respectively. In scenario two, it is approximately 11 minutes, 11 seconds, 0.1 seconds for DSIEKF 1, DSIEKF 2, and LKF, respectively.

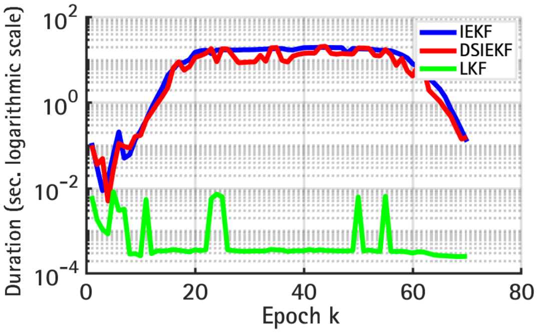

Fig. 11. Duration of the update step in scenario one.

V. Conclusions and Future Work

The core of the current paper is to adapt the idea of dual estimation to IEKF with implicit measurement equations and nonlinear geometrical constraints for the purpose of MSS georeferencing. The idea is to divide the unknown parameters into two parts, each represented by a separate vector. One of these vectors contains pose parameters of the MSS, and therefore, has a fixed size over time. The second vector contains additional geometrical information that could change in size from one epoch to the other. The new algorithm is named DSIEKF that has been applied along with IEKF and LKF on a simulated environment to georeference a UAV equipped with GNSS receiver, IMU, and a 3D laser scanner. Final results show that DSIEKF algorithm deliver similar estimations as IEKF but in a relatively shorter time. Also, in DSIEKF, planes parameters of the simulated building model are estimated more accurate than those derived by IEKF. Moreover, it is shown that by 20% subsampling of the laser scanner measurements, the computation time in DSIEKF can be significantly decreased. Because, by such subsampling, the number of laser scanner measurements in each epoch is decreased from approximately 10000 to 2000, which leads to a considerable shorter computation time. However, the 6-DOF estimations are affected, which is due to loss of measurement data. Furthermore, it is shown that although LKF is the fastest algorithm between the three considered ones, it could not be a suitable solution, since the amount of considered measurements is not enough to reach optimal state estimations. It should also be mentioned that the introduced approaches can also be applied on general applications other than georeferencing.

Further research will consider the potential effects of outliers within the laser scanner data and a suitable technique to detect them within the filtering process. Also, it will be important to see how wrong assignments of the points to the building planes can affect the results and what procedure could be taken to properly disclose these assignments and correct them. Moreover, the random selection of the measurements in a real case scenario could lead to loss of valuable data. Therefore, further studies should develop a supervised selection of the measurements within the filtering algorithm. Doing so, the number of measurements could decrease with the least possible information loss. Finally, application of the methods on real data sets rather than simulated ones as well as their improvement for real-time processing rather than post-processing are aspects of interest, which will further be investigated in the future works.

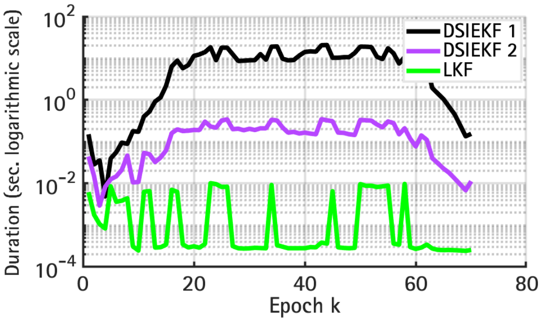

Fig. 12. Duration of the update step in scenario two.

ACKNOWLEDGMENT

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - NE 1453/5-1 and as part of the Research Training Group i.c.sens [RTG 2159]. The computations were performed by the compute cluster, which is funded by the Leibniz University of Hanover, the Lower Saxony Ministry of Science and Culture (MWK) and DFG.

REFERENCES

[1] J. Bureick, S. Vogel, I. Neumann, J. Unger, and H. Alkhatib, “Georeferencing of an unmanned aerial system by means of an iterated extended Kalman filter Using a 3D city model,” PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science, 2019, 87(5-6):229-47.

[2] S. Schön, C. Brenner, H. Alkhatib, M. Coenen, H. Dbouk, N. Garcia-Fernandez, C. Fischer, C. Heipke, K. Lohmann, I. Neumann, U. Nguyen, J.A. Paffenholz, T. Peters, F. Rottensteiner, J. Schachtschneider, M. Sester, L. Sun, S. Vogel, R. Voges, and B. Wagner, “Integrity and collaboration in dynamic sensor networks,” Sensors, 2018, 18(7):2400.

[3] S. Vogel, H. Alkhatib, and I. Neumann, “Iterated extended Kalman filter with implicit measurement equation and nonlinear constraints for information-based georeferencing,” In 2018 21st International Conference on Information Fusion (FUSION), pp. 1209-1216, IEEE.

[4] S. Schuhmacher, and J. Böhm, “Georeferencing of terrestrial laser scanner data for applications in architectural modeling,” 3D-ARCH 2005: Virtual Reconstruction and Visualization of Complex Architectures, XXXVI, PART, 2005, 5:W17.

[5] J.A. Paffenholz, “Direct geo-referencing of 3D point clouds with 3D positioning sensors,” München: Verlag der Bayerischen Akademie der Wissenschaften, Reihe C, Heft Nr. 689, 2012.

[6] S. Vogel, H. Alkhatib, and I. Neumann, “Accurate indoor georeferencing with kinematic multi sensor systems,” In 2016 International conference on indoor positioning and indoor navigation (IPIN), 2016, pp. 1-8, IEEE.

[7] J. Talaya, R. Alamus, E. Bosch, A. Serra, W. Kornus, and A. Baron, “Integration of a terrestrial laser scanner with GPS/IMU orientation sensors,” In Proceedings of the XXth ISPRS Congress 2004, Vol. 35, pp. 1049-1055.

[8] D. Dennig, J. Bureick, J. Link, D. Diener, C. Hesse, and I. Neumann, “Comprehensive and highly accurate measurements of crane runways, profiles and fastenings,” Sensors, 2017, 17(5):1118.

[9] J. Hartmann, P. Trusheim, H. Alkhatib, J.A. Paffenholz, D. Diener, and I. Neumann, “High accurate pointwise (geo-) referencing of a k-tls based multi sensor system,” ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-4 (2018), 2018, 4:81-8.

[10] T. Abmayr, F. Härtl, G. Hirzinger, D. Burschka, and C. Fröhlich, “A correlation based target finder for terrestrial laser scanning,” Journal of Applied Geodesy, 2008, 2(3):131-7.

[11] I. Elkhrachy, and W. Niemeier, “Stochastic assessment of terrestrial laser scanner,” In Proceedings of the ASPRS Annual Conference, Reno, Nevada, USA, 2006, pp. 1-5.

[12] A. Soloviev, D. Bates, and F. Van Graas, “Tight coupling of laser scanner and inertial measurements for a fully autonomous relative navigation solution,” Navigation, 2007, 54(3):189-205.

[13] J. Li-Chee-Ming, and C. Armenakis, “Determination of UAS trajectory in a known environment from FPV video,” In Proceedings of the UAV-g (Unmanned Aerial Vehicles in Geomatics) Conference, 2013.

[14] Y. Dehbi, L. Lucks, J. Behmann, L. Klingbeil, and L. Plümer, “Improving GPS trajectories using 3D city models and kinematic point clouds,” ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 2019.

[15] W.F. Denham, and S. Pines, “Sequential estimation when measurement function nonlinearity is comparable to measurement error,” AIAA journal, 1966, 4(6):1071-6.

[16] M. Tailanián, S. Paternain, R. Rosa, and R. Canetti, “Design and implementation of sensor data fusion for an autonomous quadrotor,” In 2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014, pp. 1431-1436, IEEE.

[17] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, “Collaborative monocular SLAM with multiple micro aerial vehicles,” In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 3962-3970, IEEE.

[18] T. Dang, “An iterative parameter estimation method for observation models with nonlinear constraints,” Metrology and Measurement Systems, 2008, 15(4):421-32.

[19] R. Steffen, “A robust iterative Kalman filter based On implicit measurement equations Robuster iterativer Kalman-Filter mit implizierten Beobachtungsgleichungen. Photogrammetrie-Fernerkundung-Geoinformation,” 2013(4):323-32.

[20] S. Haykin, “Kalman filtering and neural networks,” John Wiley & Sons, 2004.

[21] E.A. Wan, and A.T. Nelson, “Dual Kalman filtering methods for nonlinear prediction, smoothing and estimation,” In Advances in neural information processing systems, 1997, pp. 793-799.

[22] A. Popovici, P. Zaal, and D.M. Pool, “Dual extended Kalman filter for the identification of time-varying human manual control behaviour,” In AIAA Modeling and Simulation Technologies Conference, 2017, p. 3666.

[23] H. Khodadadi, and H. Jazayeri-Rad, ”Applying a dual extended Kalman filter for the nonlinear state and parameter estimations of a continuous stirred tank reactor,” Computers & chemical engineering, 2011, 35(11):2426-36.

[24] T.A. Wenzel, K.J. Burnham, M.V. Blundell, and R.A. Williams, ”Dual extended Kalman filter for vehicle state and parameter estimation,” Vehicle system dynamics, 2006, 44(2):153-71.

[25] D. Simon, “Kalman filtering with state constraints: a survey of linear and nonlinear algorithms,” IET Control Theory & Applications, 2010, 4(8):1303-18.

[26] D. Simon, D.L. Simon, “Kalman filtering with inequality constraints for turbofan engine health estimation,” IEE Proceedings-Control Theory and Applications, 2006, 153(3):371-8.

[27] N. Gupta, and R. Hauser,“Kalman filtering with equality and inequality state constraints,” arXiv preprint arXiv:0709.2791, 2007.

[28] V. Sircoulomb, G. Hoblos, H. Chafouk, and J. Ragot, “State estimation under nonlinear state inequality constraints. A tracking application,” In 2008 16th Mediterranean Conference on Control and Automation 2008, pp. 1669-1674, IEEE.

[29] D. Simon, and D.L. Simon, “Constrained Kalman filtering via density function truncation for turbofan engine health estimation,” International Journal of Systems Science, 2010, 41(2):159-71.

[30] L.S. Wang, Y.T. Chiang, and F.R. Chang, “Filtering method for nonlinear systems with constraints,” IEE Proceedings-Control Theory and Applications, 2002, 149(6):525-31.

[31] B.O. Teixeira, J. Chandrasekar, L.A. Tôrres, L.A. Aguirre, and D.S. Bernstein, “State estimation for linear and nonlinear equality-constrained systems,” International Journal of Control, 2009, 82(5):918-36.

[32] C. Yang, and E. Blasch, “Kalman filtering with nonlinear state constraints,” IEEE Transactions on Aerospace and Electronic Systems, 2009, 45(1):70-84.

[33] J. De Geeter, H. Van Brussel, J. De Schutter, and M. Decréton, “A smoothly constrained Kalman filter,” IEEE transactions on pattern analysis and machine intelligence, 1997, 19(10):1171-7.

[34] S. Vogel, H. Alkhatib, J. Bureick, R. Moftizadeh, and I. Neumann, “Georeferencing of laser scanner-based kinematic multi sensor systems in the context of iterated extended Kalman filters using geometrical constraints,” Sensors, 2019, 19(10):2280.

[35] J. Döllner, K. Baumann, and H. Buchholz, “Virtual 3D city models as foundation of complex urban information spaces,” CORP 2006 & Geomultimedia06, 2006.